

# Lipeärekan täytön automatisointi ja RFID-tekniikan soveltaminen lastausprosessissa

Eeli Penttilä

Opinnäytetyö  
Joulukuu 2014

Automaatiotekniikan koulutusohjelma  
Tekniikan ja liikenteen ala



JYVÄSKYLÄN AMMATTIKORKEAKOULU  
JAMK UNIVERSITY OF APPLIED SCIENCES



Tekijä(t) Penttilä, Eeli	Julkaisun laji Opinnäytetyö	Päivämäärä 10.12.2014
	Sivumäärä 49	Julkaisun kieli Suomi
		Verkojulkaisulupa myönnetty: x
Työn nimi Lipeärekan täytön automatisointi ja RFID-tekniikan soveltaminen lastausprosessissa		
Koulutusohjelma Automaatiotekniikan koulutusohjelma		
Työn ohjaaja(t) Ström, Markku		
Toimeksiantaja(t) Apex Automation Oy		
<p>Tiivistelmä</p> <p>Opinnäytetyön tarkoituksena oli luoda logiikkaohjelmus Kemira Oyj:n lipeän lastausasemalle Kemira Oyj:n vaatimusten mukaisesti. Lastausasemalla suoritetaan lipeän lastaus rekka – autoihin. Lastausasemalla suoritetuista lastauksista tulostetaan rahtikirjat sekä kirjataan lastauksen tiedot tietokantaan. Tietojen keräämistä varten työssä hyödynnetään RFID – tekniikkaa kuljettajien ja rekka- autojen tunnistuksessa.</p> <p>Lastausprosessia ohjaa Siemensin S7-300 sarjan logiikkaohjain, johon on liitetty kaksi Siemensin RFID-lukijaa sekä valvomotietokone. Logiikkaohjelmus toteutettiin Siemensin TIA-Portal ohjelmointityökalua. TIA-Portal sisältää myös WinCC valvomo-ohjelmuksen, millä toteutettiin työn valvomo-ohjelmus.</p> <p>Opinnäytetyö sisältää yleisen teoriaosuuden RFID-tekniikoista sekä ohjelmitavista logiikoista. Työosuudessa kerrotaan miten ja mitä toimilohkoja logiikkaohjelmassa on käytetty RFID-lukijoille ja prosessinohjaukseen sekä käydään läpi lastausprosessin kulkua.</p>		
Avainsanat ( <a href="#">asiasanat</a> ) RFID, ohjelmitava logiikka, TIA - Portal, Siemens		
Muut tiedot		



Author(s) Penttilä, Eeli	Type of publication Bachelor's Thesis	Date 10.12.2014
	Pages 49	Language Finnish
		Permission for web publication: x
Title Automatizing of lye truck filling and applying RFID technology in the filling process		
Degree Programme Automation Technology		
Tutor(s) Ström, Markku		
Assigned by Apex Automation Oy,		
<p>Tiivistelmä</p> <p>The Aim of the thesis project was to create PLC program to Kemira Oyj's lye loading station in accordance with the requirements of Kemira Oyj. Lye is loaded into the truck in the loading station. The Loading station prints reports when the loadings end. To collect the information for the reports, a PLC program utilized RFID technology to identify the truck and truck driver is used.</p> <p>The loading process is controlled by the Siemens S7-300 series PLC controller connected to two Siemens RFID reader as well as the PC - HMI. The Logic application was carried out by Siemens TIA - Portal programming tool. TIA - Portal also contains WinCC HMI application, which carried out the work of the PC - HMI application.</p> <p>The Thesis contains a general theoretical part of the RFID technologies and the programmable logic. The empirical part of the thesis contains information on how and what function blocks have been used for RFID readers and the process control. Additionally, the proceedings of the loading are explained in that part</p>		
Keywords RFID, programmable logic, TIA – Portal, Siemens		
Muut tiedot		

## SISÄLTÖ

1	JOHDANTO .....	6
1.1	Yritys esittelyt ja työn tarkoitus .....	6
1.1.1	Apex Automation Oy.....	6
1.1.2	Kemira Oyj .....	6
1.1.3	Opinnäytetyön tarkoitus.....	7
2	TIETOPERUSTA .....	8
2.1	RFID.....	8
2.1.1	Järjestelmä .....	9
2.1.2	Toimintaperiaate .....	10
2.1.3	Taajuusalueet .....	11
2.2	Ohjelmoitava logiikka.....	12
3	TIA-Portal .....	13
3.1	Ohjelmalohkot.....	14
4	LAITTEISTO.....	15
4.1	Siemens Simatic 315-2 PN/DP .....	15
4.2	Siemens Simatic ET 200S hajautettu I/O - asema .....	15
4.3	Siemens Simatic RF180c kommunikointimoduuli .....	16
4.4	Simatic RF260R RFID lukija.....	17
4.5	Simatic RF670R JA Simatic RF66A .....	18
4.6	Passiivitunnisteet MDS D100 ja RF620T .....	19
5	JÄRJESTELMÄKAAVIO.....	20
6	RFID-LUKIJOIDEN TESTAUS.....	21
6.1	RF670R .....	21
6.1.1	RF670R_RW toimilohko.....	22
6.2	RF180C .....	23
6.2.1	Command DB .....	25
6.2.2	Tunnisteen lukeminen.....	25
6.2.3	Tunnisteeseen kirjoittaminen.....	26
7	TOTEUTUS.....	27
7.1	Analogiamittaukset .....	28

7.2	Venttiilien ohjaukset .....	29
7.3	Pumpun ohjaukset .....	31
7.4	Automaattiohjaukset .....	33
7.5	Sekvenssi.....	34
7.6	RFID käsittelyt.....	35
7.7	Valvomosuunnittelu .....	38
7.7.1	Prosessikaavio-sivu.....	38
7.7.2	Taagit-sivu .....	39
7.7.3	Hälytykset-sivu .....	40
7.7.4	Parametrit-sivu.....	40
7.7.5	Scriptit.....	40
7.7.6	Rahtikirjan tulostus.....	41
8	IO-TESTAUS .....	41
9	KÄYTTÖÖNOTTO.....	42
10	LASTAUSTAPAHTUMA .....	44
11	POHDINTA .....	49
	LÄHTEET .....	50
	LIITTEET.....	54

- KUVIO 1. Siemens Simatic 315.2 PN/DP logiikkaohjain
- KUVIO 2. ET200S hajautettu I/O – asema.
- KUVIO 3. Simatic 180C kommunikointimoduuli profinet liitäntälohkolla
- KUVIO 4. Simatic RF260R lukija
- KUVIO 5. Simatic RF670R lukija
- KUVIO 6. Simatic RF660A antenni
- KUVIO 7. Passiivitunniste MDS D100
- KUVIO 8. Passiivitunniste RF620T
- KUVIO 9. Järjestelmäkaavio
- KUVIO 10. RF670R\_RW kommunikointi toimilohko
- KUVIO 11. RF\_VAR struktuuri
- KUVIO 12. MOBY FB – lohko
- KUVIO 13. Params\_DB
- KUVIO 14. Command datablokin sisältö
- KUVIO 15. Data Datablokki
- KUVIO 16: Analogiatulon skaalauslohko
- KUVIO 17: Pumpun P8 sulkuventtiilin ohjauslohko
- KUVIO 18: Analogiasäädin
- KUVIO 19: Pumpun kierrosnopeuden syötönesto
- KUVIO 20: P8 sulkuventtiilin automaattiajo Auki
- KUVIO 21: Sekvenssikaavio
- KUVIO 21: ID-numero vertailu henkilötunniste
- KUVIO 22: Rekka- auton tunnisteen vertailu
- KUVIO 23: Venttiilin ohjausikkuna

KUVIO 24. Valvomonäyttö

KUVIO 25. Valvomonäyttö, asiakaslista

KUVIO 26. Valvomonäyttö, rahtikirjan pohja

KUVIO 27: Valvomonäyttö, uloskirjautuminen lastauspaikalta

## LYHENTEET JA KÄSITTEET

LF – Low frequency

HF – High frequency

VHF – Very high frequency

UHF – Ultra high frequency

SHF – Super high frequency

TCP – Transmission Control Protocol

mA – Milliampeeri

XML – Extensible Markup Language



# 1 JOHDANTO

## 1.1 Yritys esittelyt ja työn tarkoitus

### 1.1.1 Apex Automation Oy

Opinnäytetyön toimeksiantajana oli Kokkolalainen insinööritoimisto Apex Automation Oy. Yritys on perustettu vuonna 1993 ja se on erikoistunut automaatio- ja sähkösuunnitteluun. 20 vuoden toiminta-aikanaan Apex Automation on hankkinut laajan asiakaskunnan koko Suomen alueelta sekä on merkittävin alansa toimija Kokkolan lähialueella. Apex Automation työllistää tällä hetkellä noin 50 henkilöä. (Apex Automation Oy, Kotisivut)

Apex Automation Oy tarjoaa suunnittelupalveluja ja automaatiojärjestelmiä "Avaimet käteen" -periaatteella tai pienempinä osatoimituksina. (Apex Automation Oy, Kotisivut)

### 1.1.2 Kemira Oyj

Kemira Oyj on kansainvälinen vesikemian yhtiö. Kemiran keskeisiä asiakkaita ovat runsaasti vettä kuluttavat teollisuuden alat. Se tarjoaa veden laadun ja määrän hallintaratkaisuja, joilla lisätään asiakkaiden energia-, vesi- ja raaka-ainetehokkuutta. (Konserni, Kemiran kotisivut)

Kemira on organisoitu neljään segmenttiin. Paper-segmentti tarjoaa tuotteitaan ja palveluitaan massa- ja paperiteollisuudelle, Municipal & Industriasegmentti kunnalliseen ja teolliseen vedenpuhdistukseen, Oil & Mining-segmentti tarjoaa erotus- ja prosessiratkaisuja öljy- ja kaivosteollisuudelle ja

ChemSolutions palvelee muun muassa rehu- ja kemianteollisuutta. (Konserni, Kemiran kotisivut)

### 1.1.3 Opinnäytetyön tarkoitus

Opinnäytetyö tehtiin Kemiran lipeän lastausasemalle, joka sijaitsee Porin Tahkoluodossa. Lastausasemalla suoritetaan lipeänlastaus rekka-autoihin. Kemira oli rakentanut uuden lastausaseman vanhan lastausaseman viereen ja tälle uudelle asemalle oli tarkoitus suunnitella lipeän lastausjärjestelmä Siemensin TIA-Portal ohjelmointityökalulla.

Vanha lastausjärjestelmä ei sisältänyt lipeän määrän mittausta. Rekka-autojen täytyi aina käydä vaa’alla ennen ja jälkeen lastauksen punnitsemassa auto, josta heille tulostui rahtikirja. Kemira halusi, että vaa’alla käynnistä voitaisiin luopua ja, että lastausjärjestelmä tulostaisi kuljettajille tarvittavat rahtikirjat. Jotta rahtikirjan tulostus olisi mahdollista, täytyi lastaustapahtumasta kerätä tiedot tulostettaviin rahtikirjoihin. Tiedot rahtikirjoihin kerätään MYSQL-tietokannan, RFID-tekniikan sekä Ontec Oy:n sertifioitun mittausjärjestelmän avulla.

Työn tavoitteena oli toteuttaa toimiva lastausjärjestelmä, joka vastaan Kemiran asettamia ehtoja. Lastausjärjestelmän piti olla selkeä ja niin helppokäyttöinen, että kuljettajat oppivat sen toiminnan nopeasti.

Työ käsittelee RFID-tekniikan soveltamista kyseissä järjestelmässä sekä logiikka- ja valvomosuunnittelua.

## 2 TIETOPERUSTA

### 2.1 RFID

RFID on lyhenne sanoista radio frequency identification, joka tarkoittaa radio-taajuista tunnistusta. RFID-tekniikkaa käytetään tuotteiden ja asioiden tunnistamiseen, havainnointiin ja yksilöintiin. Toiminta perustuu tiedon tallentamiseen RFID-tunnisteeseen ja sen langattomaan lukemiseen RFID-lukijalla radioaaltojen avulla. (RFID – tietoutta, RFIDLab kotisivut) Tekniikan etuna on mahdollisuus lukea iso joukko tunnisteita kerralla ja tunnistaminen onnistuu erilaisten materiaalien läpi (RFID - etätunnistusteknologia 2012, 110).

RFID-tekniikka sai alkunsa, kun vuonna 1935 Sir Robert Alexander Watson-Wattin keksimää tutkaa käytettiin toisen maailmansodan aikana varoittamaan lähestyvistä lentokoneista. Tutkan ainoa ongelma oli, että se pystyi vain erottamaan ilmassa lentävän lentokoneen, eikä tunnistamaan kenen lentokone ilmassa oli. Britanniassa keksittiin lisätä lentokoneisiin lähetin, jonka avulla pystyttiin erottamaan brittikoneet saksalaisista. Järjestelmää kutsuttiin nimeltä IFF (identify Friend or Foe) ja se oli maailman ensimmäinen RFID-järjestelmä. (RFID – tekniikan historia, RFIDLab kotisivut)

50-luvulla tekniikkaa alettiin hyödyntää varashälyttimissä ja sama tekniikka on käytössä hälyttimissä nykyäänkin. Ensimmäisiä kaupallisia sovelluksia otettiin käyttöön 1980-luvun puolivälissä tietulleissa. (RFID - etätunnistusteknologia 2012, 110)

Nykyään RFID-tekniikkaa hyödynnetään laajasti. Lääketeollisuudessa lääkevalmistajat ja sen asiakkaat hyödyntävät RFID-tekniikkaa lääkkeiden aitouden

tunnistukseen liimaamalla lääkepakkauksiin RFID-tunnisteen. Sairaaloissa potilaille voidaan antaa ranneke, joka sisältää RFID-tunnisteen, jonka lukemalla saadaan tiedot potilaalle määrätystä lääkkeistä. (Helpotusta sosiaali ja terveystalveluiden arkeen rfid:n avulla, , RFIDLab kotisivut)

Kirjastoissa kirjoissa olevat viivakoodit voidaan korvata RFID-tunnisteella. Aineiston kierto nopeutuu ja tarkkuus sekä aineiston hallinta hyllyssä paranuu. Tunniste suojaa myös aineistoa hävikiltä. (RFID, Mikroväylä kotisivut)

Kaupanalalla tavaroiden liikettä voidaan seurata paremmin ja tarkemmin hyödyntäen RFID-tekniikkaa. Tavaroiden automaattisella tunnistamisella saadaan toiminnalle parempi ennustettavuus sekä kykyä reagoida muutoksiin paremmin (RFID kaupanalalla, RFIDLab kotisivut). RFID-tekniikkaan siirtyneet yritykset ovat tehostaneet toimitusketjujaan ja kaupankäyntiä. Vastaanotto nopeutuu 80 % ja kaupan inventointi voidaan suorittaa minuuteissa, päivien sijaan. (Tehokkuutta toimitusketjun hallintaan ja kaupankäyntiin RFID – tekniikalla, Jukka Wallinheimo)

### 2.1.1 Järjestelmä

RFID- järjestelmät koostuvat lukulaitteesta, tunnistesta sekä taustajärjestelmästä joka käsittelee tunnistesta luettua tietoa. Järjestelmän idea on yksinkertainen. RFID tunniste asetetaan haluttuun kohteeseen ja lukijan avulla siihen joko, kirjoitetaan tai siitä luetaan tietoa. Luettua tietoa voidaan sen jälkeen hyödyntää taustajärjestelmässä. RFID-järjestelmä toimii samalla periaatteella kuin viivakoodilla toteutettu järjestelmä. RFID:llä toteutettu järjestelmä eroaa viivakoodilla toteutetusta siten, että tunnistus voi tapahtua ilman suoraa katsekontaktia tunnisteseen. Tämän lisäksi RFID-tunnisteen sisältöä voidaan

muokata jälkikäteen, toisin kuin viivakoodia, joka on tulostuksen jälkeen muuttumaton. RFID - tunnistet myös ovat paljon pitkäikäisempiä kuin viivakoodit. (RFID – tietoutta, RFIDLab kotisivut)

### 2.1.2 Toimintaperiaate

RFID-tunnisteita on kolmea eri tyyppiä; Passiivi-, puoli-passiivi- ja aktiivitunniste.

Passiivitunnisteet eivät sisällä omaa virtalähdettä vaan tunnisteen käyttö sähkö indusoituu siihen radiosignaalista.

Passiivitunnisteet voidaan jakaa kahteen ryhmään: matalan (LF) ja korkean taajuuden (HF) tunnistet sekä korkeampien taajuuksien tunnistet (VHF, UHF ja SHF). (RFID – tekniikan fysiikkaiset perusteet, RFIDLab kotisivut)

LF ja HF taajuusalueilla keskustelu tapahtuu induktiivisen kytkennän avulla. Tunnisteessa sekä lukijassa on kuparisia silmukoita, jotka muodostavat käämin, joka toimii ns. antennina. Kun lukija johtaa vaihtovirtaa käämiinsä, se luo oskilloivan magneettikentän. Tämä magneettikenttä indusoi vaihtovirran tunnisteen silmukkaan, jos se sijaitsee tarpeeksi lähellä lukijaa. Tällöin tunnisteteessa oleva siru saa virtansa ja sirun muistissa oleva dataa käytetään moduloimaan tunnisteen käämin virtaa, mikä taas näkyy magneettikentän yli lukijan silmukan jännitteessä. Tunnisteen asennolla on merkitystä induktiivisesta kytkennästä johtuen, mutta käytännössä tunniste voi olla monissa eri asennoissa tekniikan toimiessa oikein. (RFID – tekniikan fysiikkaiset perusteet, RFIDLab kotisivut)

VHF, UHF- ja mikroaaltotaajuusalueilla tunniste ja lukija keskustelevat keskenään radioaaltoja välittämällä, samaan tapaan kuin matkapuhelimet tai ra-

diot. Lukija lähettää antenninsa kautta radioaaltoja, ja tunnisteen dipoliantenni vastaanottaa aallot ja heijastaa niitä takaisin sisältäen sirun tiedot. Tunniste voi välittää heijastetussa signaalissa tiedot lukijalle monella eri tavalla, kuten nostamalla heijastuneen signaalin amplitudia, siirtämällä heijastuneen signaalin vaihetta tai muuntamalla heijastuneen signaalin taajuutta. (RFID – tekniikan fysikaaliset perusteet, RFIDLab kotisivut)

Passiivitunniste voi olla kokonsa puolesta hyvinkin pieni, koska se ei tarvitse virtalähdettä. Se on myös aktiivitunnistetta pitkäkestoisempi ja tämän vuoksi suurin osa käytössä olevista tunnisteista on passiivia.

Aktiivitunniste sisältää oman pariston tai akun, jolloin tunniste ja lukija keskustelevat kuin kaksi radiota. Tämän ansiosta sillä on myös pidempi kantomatkä ja suurempi muisti kuin passiivitunnisteilla. Aktiivitunnisteet toimivat aina joko VHF- tai UHF-alueella ja niiden kantomatkä voi olla jopa satoja metrejä. (RFID – tekniikan fysikaaliset perusteet, RFIDLab kotisivut)

Puoli-passiivitunnisteet sisältävät pariston tai akun, mutta käyttävät sitä ylläpitämään omaa virtapiiriään. Kommunikointiin lukijan kanssa tunniste saa virtansa lukijan radiosignaalista.

### **2.1.3 Taajuusalueet**

RFID tekniikassa käytetään yleisesti neljää eri taajuusaluetta.

Matalan taajuuden (LF) tunnistet toimivat 125 kHz – 134 kHz alueella. Tämän taajuusalueen tunnistetä käytetään yleisesti eläinten tunnistukseen ja joissakin kulunvalvontajärjestelmissä. Korkean taajuuden (HF) tunnisteen standarditaajuus on 13,56 MHz. Kyseisen taajuuden tunnisteen lukuetaisyys optimiolosuhteissa on noin 1,5m ja sitä käytetään kulunvalvonnassa. UHF-

alueen tunnistet käyttötaajuus Euroopassa on 865–868 MHz ympärillä ja Yhdysvalloissa 902–928 MHz. Sitä käytetään yleisesti logistiikkasovelluksissa. Mikroaaltoalueen yleisin taajuus on 2,4GHz. Tunnetuimpia sovelluksia on automaattinen tunnistus tietulleissa. (RFID – tekniikan käyttämät taajuusalueet, RFIDLab kotisivut)

## 2.2 Ohjelmoitava logiikka

Ohjelmoitava logiikka keksittiin korvaamaan relepiirejä koneohjauksissa. Yhdellä logiikalla voitiin korvata jopa tuhansia aiemmin käytettyjä releitä tai ajastimia. Se on digitaalisesti toimiva elektroninen laite, joka käyttää ohjelmoitavaa muistia, johon voidaan ohjelmoida erilaisia toimintoja (HOME, PLCTutor kotisivut). Ohjelmoitavassa logiikassa on tulo- ja lähtöportteja, joihin kaikki kenttälaitteet on kytketty. Logiikka ohjaa toimilaitteita tehdyn ohjelman ja sensoreiden antamien tietojen mukaisesti. (Keinänen, Kärkkäinen, Lähetkangas, Sumujärvi 2007, s.212)

Logiikka suorittaa siihen tehtyä ohjelmaa kiertävästi. Ohjelmakierron alussa logiikka lukee tulojen ja lähtöjen tilat omalle muistialueelle. Tämän jälkeen se etenee ohjelmarivi kerrallaan ohjelmaa läpi. Tulos käsitellään ja toteutetaan siinä järjestyksessä, kun ohjelmaa luetaan. Ohjelmassa olevat käskyt toteutetaan lähdöille vasta, kun koko ohjelmakierros on luettu. (Keinänen, Kärkkäinen, Lähetkangas, Sumujärvi 2007, s.223)

Logiikoiden ohjelmointikielen peruselementit muodostuvat logiikkaporteista ja muista käskysanoista, joilla käsitellään esim. ajastimia, laskureita tai apumuisteja. Nämä ovat standardisoimattomia ja ne vaihtelevat valmistajasta riippuen. (Keinänen, Kärkkäinen, Lähetkangas, Sumujärvi 2007 s.223)

Logiikoissa on yleisesti käytössä viisi eri standardoitua kieltä. FBD (Function Block Diagram) eli toimilohkokaavio rakentuu toisiinsa johdetuista toimilohkoista ja se muistuttaa ulkonäöltään mikropiireillä toteutettua ohjainkortin kaaviota. LD (Ladder Diagram) eli tikapuukaavio, joka näyttää samalta kuin sähköpiirikaavio. LD ohjelmointikielen käyttö on erittäin yleistä. ST (Structure Text) eli rakenteellinen teksti, joka muistuttaa Pascal ohjelmointikieltä. IL (Instruction List) eli käskylista, joka muistuttaa assembler ohjelmointikieltä. ST ja IL ohjelmointikielet ovat niin sanottuja rakenteellisia tekstieditoreja. Ohjelmointitapa sisältää yksinkertaisia tekstimuotoisia komentoja. Kielen lausekkeet perustuvat IF, THEN, ELSE rakenteeseen. SFC (Sequential Function Chart) on tarkoitettu ylemmätason ohjelmointiin, kuten järjestelmän toimintamoodin valitsemiseen. Se ei kielenä tarjoa kovinkaan monipuolisia ominaisuuksia itse logiikan ohjelmointiin, vaan se on vain runko varsinaiselle logiikkaohjelmalle. (Keinänen, Kärkkäinen, Lähetkangas, Sumujärvi 2007 s.224–225)

### 3 TIA-Portal

Siemens TIA-Portal -lyhenne tulee englannin sanoista totally integrated automation. TIA-Portaaliin on integroitu kaikki erillisohjelmat yhden ohjelman alle. Samalla työkalulla voidaan ohjelmoida logiikat, käyttöliittymät, turvaratkaisut sekä määritellä taajuusmuuttajat. Tämä ominaisuus erottaa TIA-portaalin edeltäjästään SIMATIC Managerista. TIA-Portaalin käyttöliittymän käyttäjäystävällisyys soveltuu erinomaisesti niin vasta-alkajalle kuin kokeneelle suunnittelijalle. TIA-Portal pitää sisällään Step7-työkalun, WinCC-työkalun, ja StartDrive-työkalun, joiden käyttöliittymää on kehitetty integroinnin yhteydessä. TIA-Portaalilla on mahdollista ohjelmoida kaikkia S7-sarjan logiikoita sekä uusia S7-1200-ja S7-1500-sarjan logiikoita. (TIA-Portal Suunnittelun uusi aikakausi).



### 3.1 Ohjelmalohkot

TIA-Portal sisältää ohjelmalohkoja joiden avulla ohjelma rakennetaan. OB eli organisaatioyksikkö on käyttöliittymä käyttöjärjestelmän ja käyttäjän luoman ohjelman välillä. Käyttöjärjestelmä kutsuu organisaatioyksikköä ja ne ohjaavat logiikan käynnistystoimenpiteitä, syklisen ohjelman prosessointia, keskeytyksiä ja virheiden käsittelyä. Ohjelmaan voidaan luoda useita OB-lohkoja joiden käsittelyjärjestys määräytyy OB numeron mukaan. (Programming Guideline 2014, 28-29)

FC eli toiminnot ovat lohkoja ilman syklisiä tietojen tallennusta. Kyseisen lohkoille täytyy aina syöttää tietoa, kun niitä kutsutaan ohjelmassa sillä lohkon sisäiset ja ulostulo muuttujat menettävät tietonsa lohkon läpikäynnin jälkeen. (Programming Guideline 2014, 30-31)

FB eli toimintalohko on lohko, joka sisältää syklisen tietojen tallennuksen. Lohkot kutsuvat omaa instanssidatalohkoa minne tiedot tallentuvat pysyvästi. Instanssidatalohkon pysyvää muistia ovat lohkon sisällä olevat input, output, inout ja static muuttujat. Lohkon sisältämä temp muuttuja taas menettää tietonsa ohjelmakierron loputtua. (Programming Guideline 2014, 32)

FB lohkoja voidaan kutsua useaan kertaan eri ohjelmaosioissa. Tämä tekee toistuvien ohjelmanosien ohjelmoinnista helpompaa.

DB eli datalohko on muistialue minne tallennetaan tietoa. Datalohkoja on kah- ta eri tyyppiä. Global DB on kaikkein lohkojen käytössä oleva muistialue, kun taas instanssidata lohko on sidottuna sitä kutsuvaan FB-lohkoon. (Programming Guideline 2014, 33-36)

## 4 LAITTEISTO

### 4.1 Siemens Simatic 315-2 PN/DP

Projektissa käytimme Siemens Simatics 315-2 PN/DP logiikkaohjainta. Se on Siemensin valmistama S7-300 tuoteperheeseen kuuluva logiikkaohjain, jonka yleisimmät käyttökohteet ovat erilaisista prosessi- ja kappaletavarateollisuuden ohjauksista aina yksittäisten koneiden ohjauksiin. (SIMATIC S7-300, Siemens kotisivut)

Kyseisen logiikkaohjaimen valintaperusteet olivat sen muistin koko sekä profinet-väyläliitäntä. Se sisältää kaksinkertaisen määrän muistia aiemmin käytössä olleeseen ET-200S logiikkaohjaimeen nähden.



KUVIO 1. Siemens Simatic 315-2 PN/DP logiikkaohjain. (Siemens Support)

### 4.2 Siemens Simatic ET 200S hajautettu I/O - asema

Hajautettua I/O - asemaa käytetään yleensä silloin, kun halutaan viedä tulo- ja lähtöpiirit lähemmäksi toimilaitteita. Täten kaapelointi helpottuu ja säästetään kustannuksissa.

Alun perin projektissa logiikkaohjaimena käytettiin ET 200S hajautettua I/O – asemaa, johon oli integroituna logiikkaohjain. Sen muisti kuitenkin täyttyi ja se korvattiin edellä mainitulla S7-300 sarjan logiikkaohjaimella. Ohjaukset ja mittaukset oli kuitenkin jo kytketty ET 200S ohjaimen kortteihin, joten se jätettiin hajautetuksi I/O – asemaksi prosessiin vaihtamalla ET200S pelkäksi interface moduuliksi.



KUVIO 2: ET200S hajautettu I/O – asema. (Siemens tuoteuutinen, 2010)

### 4.3 Siemens Simatic RF180c kommunikointimoduuli

RF180C on Siemensin valmistama kommunikointimoduuli RFID-lukijoita varten. Sen avulla saadaan liitettyä RFID lukijat profinet-väylään. Profinet-väylään voidaan liittyä normaalilla RJ45 liittimellä tai M12 liitäntä holkillla. Molemmille liitäntätavoille on erilliset liitäntälohkot. Yhteen kommunikointimoduuliin on mahdollista liittää kaksi RFID-lukijaa.

Kommunikointimoduuli on yhteen sopiva seuraavien RFID tuoteperheiden kanssa: RF200, RF300, RF600 (RF630 ja RF630), MOBY D, MOBY U, MOBY E ja MOBY I.



KUVIO 3. Simatic RF180C kommunikointimoduuli profinet liitäntälohkolla (Aksprom)

#### 4.4 Simatic RF260R RFID lukija

SIMATICS RF260R kuuluu uuteen RF200 RFID-HF tuoteperheeseen, jotka ovat lyhyen ja keskimatkan lukijoita. RF260R lukijaan on integroituna antenni ja se käyttää HF standarditaajuutta, joka on 13,56MHz. Lukijan tunnistusetäisyys on noin 130mm. Lukijassa on kolmivärinen led valo, joka kertoo lukijan tilasta. Lukijalla toteutettiin työssä kuljettajien tunnistus.



KUVIO 4. Simatic RF260R RFID - lukija (Siemens Support, 2012)

## 4.5 Simatic RF670R JA Simatic RF66A

RF600 tuoteperheeseen kuuluva RF670R, joka on passiivinen pitkän matkan RFID - lukija. RF670R toimii UHF taajuusalueella ja se tarvitsee toimiakseen ulkopuolisen antennin. Lukijaan voidaan liittää yhteensä 4 kappaletta antennia ja yhden antennin lukuetaisyys voi olla maksimissaan 8 metriä riippuen antennin ja luettavan tunnisteen tyypistä. Lukija liitetään logiikkaan ethernet kaapelilla ja kommunikointi pohjautuu TCP/IP protokollaan ja XML rajapintaan. Lukijan mukana tulee RF-Manager Basic ohjelmisto, jonka avulla lukija konfiguroidaan. Ohjeet laitteen konfigurointiin löytyy liitteestä 1.



KUVIO 5. Simatic RF670R RFID – lukija (Siemens Parts Online)

Simatic RF660A on RF600 sarjan lukijoihin soveltuva antenni, jonka avulla saadaan luettua RFID signaaleja UHF taajuusalueella. Antenni liitetään lukijaan sille tarkoitettulla kaapelilla.



KUVIO 6. RF670R RFID - lukijan antenni (TPAutomation)

Lukijan avulla toteutettiin työssä rekka-autojen tunnistus. Lukijaan liitettyjen kahden antennin avulla luku etäisyys oli noin 4 metriä.

#### 4.6 Passiivitunnisteet MDS D100 ja RF620T

MDS D100 on pankkikortin näköinen tunniste, joka sisältää 128 tavua muistia. Tunnisteen toimintataajuus on 13.56MHz ja lukuetaisyys 650mm. Tunniste soveltuu hyvin henkilötunnistukseen lyhyen lukuetaisyytensä takia. Tunniste on yhteensopiva RF200 ja RF300 - sarjan lukijoiden kanssa. Korttia käytettiin työssä kuljettajien tunnistukseen.



KUVIO 7. Siemens Simatic MDS D100 passiivitunniste (Siemens Support )

RF620T on RF600 sarjan lukijoille suunniteltu passiivitunniste, joka sisältää 64 tavua muistia. Tunnisteen toimintataajuus 860 - 960MHz ja lukuetaisyys parhaimmillaan 8 metriä. Kyseistä tunnistetta käytettiin työssä rekka-autojen tunnistukseen.

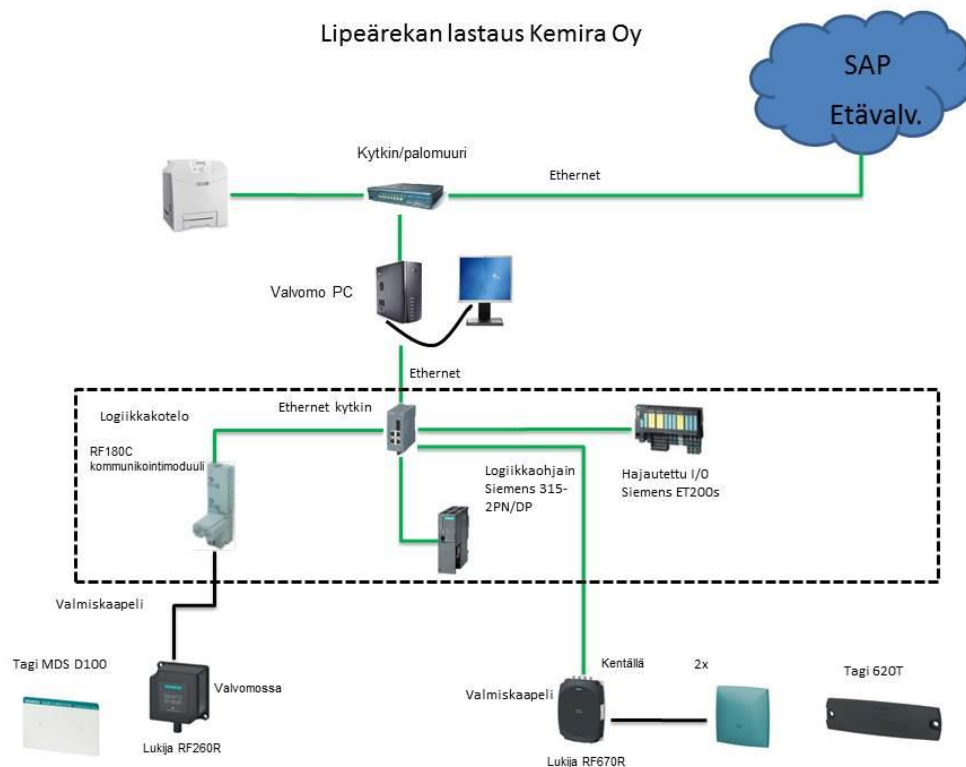


KUVIO 8. Siemens Simatic RF620T passiivitunniste (Siemens Industry mall)

## 5 JÄRJESTELMÄKAAVIO

Lastausaseman ohjausjärjestelmä koostuu järjestelmäkaaviossa (kuvio 9) olevista laitteista.

Lastausjärjestelmä sisältää valvomo tietokoneen. Siihen on asennettu Siemens Simatic WinCC valvomosovellus, jonka avulla lastausta hallitaan. Tietokoneeseen on asennettuna myös UltraVNC sovellus, jonka avulla tietokoneeseen päästään etäyhteydellä kiinni ongelmatapauksissa. Valvomotietokone sisältää kaksi verkkokorttia. Toinen verkkokortti on yhteydessä Kemiran verkkoon ja toinen ns. tehdasverkkoon, jonka alla on lastausta ohjaava logiikka, hajautettu I/O-asema ja RFID - lukijat. Tällöin saadaan lastausta ohjaavat laitteet erotettua Kemiran omasta verkosta. Tämä helpottaa ip-osoite asetusten kanssa, koska voidaan itse määrittää mitä ip-osoitteita käytetään tehdasverkossa.



KUVIO 9: Järjestelmäkaavio

## 6 RFID-LUKIJOIDEN TESTAUS

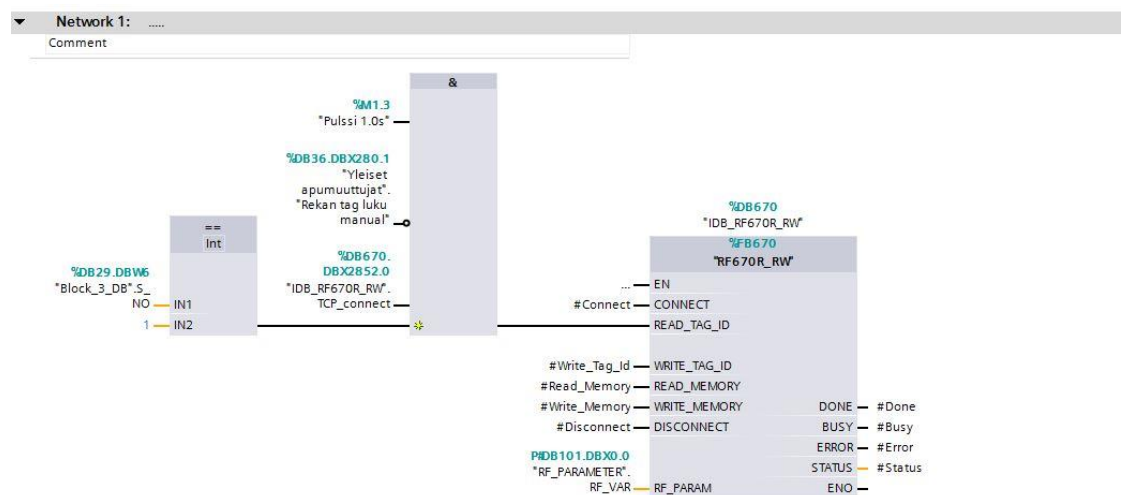
RFID-lukijoiden testaus toteutettiin Apex Automation Oy:n toimistolla. Ensimmäiseksi otettiin käsittelyyn RF670R lukija.

### 6.1 RF670R

Ensimmäiseksi lukija parametroidiin RF-Manager Basic ohjelmalla. Ohjelmalla lukijalle asetetaan ip-osoite, käytössä olevien antennien määrä sekä maa, jossa lukijaa käytetään. Parametrointi ohjeet löytyvät liitteestä 1.

Seuraavaksi selvitettiin, kuinka Siemensin logiikalla saadaan tiedonsiirto toimimaan RF670R-lukijan kanssa. Lukijaa ei lisätä logiikan hardware konfiguraatioon, sillä logiikan ja lukijan väliseen kommunikointiin käytetään XML-rajapintaa.

RF670R-lukijan ja logiikan väliseen tiedonsiirtoon käytettiin Siemensiltä saatua FB670 toimilohkoa. Lohkon avulla luodaan yhteys RF670R-lukijaan sekä sen avulla voidaan lukea tai kirjoittaa tunnisteisiin tietoa.



KUVIO 10. RF670R\_RW kommunikointi toimilohko



### 6.1.1 RF670R\_RW toimilohko

CONNECTION-nastan nouseva reuna lähettää logiikalta lukijalle yhteyden avauspyynnön. Jos yhteyden luonti onnistuu, DONE-nasta menee tilaan "1" ja lukijan ledi syttyy vihreäksi.

DISCONNECT-nasta lähettää lukijalle yhteyden katkaisupyynnön.

READ\_TAG\_ID-nasta lähettää lukijalle tunnisteen lukupyyntö. Kun nastaan tulee nouseva reuna, lukija ottaa antennien lukukentässä olevan tunnisteen ID-numeron talteen ja kirjoittaa sen RF\_PARAM nastassa olevaan "RF\_PARAMETER". RF\_VAR struktuurin sisällä olevaan TAG\_ID muuttujaan. Jos luku onnistuu, DONE-nasta menee tilaan "1".

WRITE\_TAG\_ID-nastan nouseva reuna kirjoittaa tunnistelle uuden ID-numeron. Tunnisteen uudelleen kirjoittaminen vaatii kuitenkin, että tunnisteen ID-numero on ensiksi luettu TAG\_ID muuttujaan. Tämän jälkeen RF\_PARAM new\_ID muuttujaan kirjoitetaan uusi ID-numero.

RF\_PARAM-nastaan tuodaan RF\_VAR-tyyppinen struktuuri, joka sisältää muuttujia joita tarvitaan TCP yhteyttä varten sekä liikkuvan datan tiedonkäsittelyyn (KUVIO 10).

Alle on lueteltu tärkeimmät muuttujat RF\_VAR struktuurista.

IP\_ADR – Lukijalle asetettu ip-osoite.

DEVICE\_ID – profinet-väylän slotti-osoite logiikassa. Osoitteen voi tarkistaa logiikan hardware konfiguraationista.

CON\_ID – yksilöllinen yhteys ID TCP yhteyttä varten.

LOCAL\_PORT – käytössä oleva portti.

REC\_Data – Luetun tunnisteiden ID-numero.

new\_ID – Tunnisteiden uusi ID-numero.

Source – Lähde, joka on määritelty lukijalle RF-Manager Basic ohjelmalla.

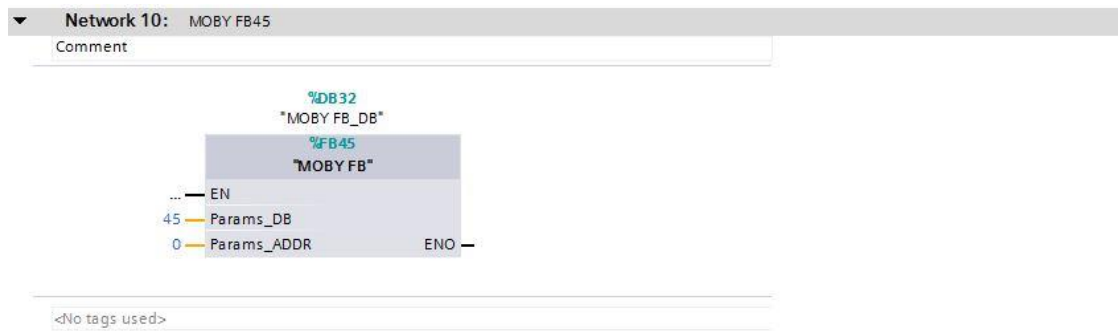
	Name	Data type	Offset	Start value	Retain	Visible in ...	Comment
1	Static						
2	RF_VAR	*RF_VAR*	0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
3	IP_ADR	DWord	0.0	16#0A14_1E23	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
4	DEVICE_ID	Byte	4.0	B#16#2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
5	CON_ID	Word	6.0	W#16#60	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
6	LOCAL_PORT	Word	8.0	W#16#7D0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
7	TAG_ID	String[24]	10.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
8	new_ID	String[24]	36.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
9	Source	String[254]	62.0	'Source_1'	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
10	Value_bank	Int	318.0	3	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
11	value_startAddress	Int	320.0	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
12	value_dataLength	Int	322.0	2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
13	Duration	Int	324.0	50	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
14	WRITE_DATA	Array[1 .. 128] Of ...	326.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
15	TAG_IDs	Int	454.0	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
16	LNG	Int	456.0	6	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
17	REC_DATA	Array[1 .. 150] Of ...	458.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

KUVIO 11. RF\_VAR struktuuri

## 6.2 RF180C

Ensimmäiseksi lukija lisättiin logiikan hardware konfiguraatioon. Samalla sille määriteltiin tulo- ja lähtöosoitteet sekä mykistettiin toisen kanavan virheilmoituksen lähetys logiikalle, koska käytössä vain yksi lukija. Tulo- ja lähtöosoitteet täytyi asettaa samoiksi. Kun konfiguratiossa saatiin valmiiksi, alettiin tutkia Siemensiltä saatua MOBY45-toimilohkoa.

Kyseistä lohkoa käytetään logiikan ja RF180C lukijan väliseen tiedonsiirtoon. Kyseinen lohko sisältää kaksi tulonastaa. Params\_DB tuloon annetaan datalokin numero, josta lohko käy lukemassa parametrit. Params\_ADDR tuloon annetaan osoitin numero, joka määrittää mistä muuttujasta Params\_DB:tä parametreja aletaan lukea.



KUVIO 12. MOBY FB – lohko

Kuviossa 13 on nähtävillä Params\_DB:n sisältöä. Ottaessa lukijaa käyttöön pitää ASM\_address-, ASM\_channel-, command\_DB\_number- ja command\_DB\_address-muuttujiin tehdä muutoksia. ASM\_address-muuttujaan tulee sama lukuarvo, kuin mikä on määritelty hardware konfiguraatiossa lukijan tulo- ja lähtöosoitteeksi. ASM\_channel-muuttujaan tulee määrittää lukijoiden määrä, joka on tässä tapauksessa yksi. Command\_DB\_number-muuttujaan annetaan komento-datablokin numero. Kyseinen datablokin avulla annetaan lukijalle luku- tai kirjoituspyyntö. Command\_DB\_address-muuttujaan annetaan osoitin mistä kohtaa Command\_DB:tä aloitetaan sen lukeminen. Kun olin saanut yllämainitut muuttujat kohdalleen ja ohjelman laaduttua logiikkaan, pääsin testaamaan lukijan toimivuutta.

MOBY DB						
	Name	Data type	Offset	Start value	Retain	Visible in ...
1	Static					
2	SLG1	*MOBY Para...	0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3	ASM_address	Int	0.0	280	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
4	ASM_channel	Int	2.0	1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
5	command_DB_number	Int	4.0	47	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
6	command_DB_address	Int	6.0	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
7	MDS_control	Byte	8.0	B#16#1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
8	ECC_mode	Bool	9.0	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
9	RESET_long	Bool	9.1	TRUE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
10	MOBY_mode	Byte	10.0	B#16#5	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
11	scanning_time	Byte	11.0	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
12	option_1	Byte	12.0	B#16#2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

KUVIO 13. Params\_DB

Ensimmäiseksi lukijalle annetaan "init\_run"-käsky, joka suorittaa kommunikointimoduulille alustuksen ja asettaa params\_DB parametrit kommunikointimoduuliin. Jos lukijassa oleva ledi palaa vihreänä "init\_run"-käskyn jälkeen, yhteys kommunikointimoduulin ja lukijan välillä on kunnossa. "init\_run"-käsky asetetaan OB 100-lohkoon, koska kyseinen lohko suoritetaan, kun logiikka käynnistetään.

### 6.2.1 Command DB

Command\_DB on datablokki, joka määriteltiin Moby DB:n command\_DB\_number-muuttujaan. Datablokki sisältää command-muuttujan, minkä avulla lukija asetetaan kirjoitus- tai lukutilaan. DAT\_DB\_number-muuttujaan määritellään datablokin numero, mihin tallennetaan tunnistesta luettu id-numero ja DAT\_DB\_address-muuttujaan, mihin sanaan datablokin sisällä luettu id-numero tallennetaan.

	Name	Data type	Offset	Start value	Retain	Visible in ...	Comment
1	Static						
2	Kanal_1_Befehl	Array [1 .. 5] Of "M..	0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
3	Kanal_1_Befehl[1]	"MOBY CMD_e"			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
4	command	Byte		B#16#2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
5	sub_command	Byte		0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
6	length	Int		1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
7	address_MDS	Word		0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
8	DAT_DB_number	Int		48	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
9	DAT_DB_address	Int		0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

KUVIO 14. Command datablokin sisältö.

### 6.2.2 Tunnisteen lukeminen

Tunnisteen lukemista varten täytyy lukija olla lukutilassa. Lukutila asetetaan päälle kirjoittamalla aiemmin määritellyyn Command\_DB:n command-muuttujaan heksana luku 2. Kun tunnistetta luetaan, täytyy lukijan ensin havaita, että sen lukukentässä on tunniste. Params\_DB sisältää muuttujan, joka ilmaisee asian. Kun tunniste on antennin lukukentässä, annetaan lukijalle

käsky lukea tunniste. Luettu ID-numero tallentuu Data nimisen datablokin Daten[1]-muuttujaan.

### 6.2.3 Tunnisteeseen kirjoittaminen

Kun tunnisteelle halutaan kirjoittaa uusi ID-numero, täytyy lukija asettaa kirjoitustilaan. Kirjoitus tilaan päästään kirjoittamalla command-muuttujaan heksana luku 1. Tämän jälkeen kirjoitetaan Data datablokin Daten[1] - muuttujaan uusi ID - numero ja käytetään tunnistetta antennin lähettyvillä. Kun antenni tunnistaa tunnisteiden lukukentässä, siirtää lukija Daten[1] muuttujaan määritellyn ID-numeron tunnisteeseen.

Data						
	Name	Data type	Offset	Start value	Retain	Visible in ...
1	Static					
2	▼ Daten	Array [1 .. 64] Of By...	0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3	■ Daten[1]	Byte		0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
4	■ Daten[2]	Byte		0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
5	■ Daten[3]	Byte		0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
6	■ Daten[4]	Byte		0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

KUVIO 15. Data Datablokki

## 7 TOTEUTUS

Projekti aloitettiin tutustumalla lastauksen toimintakuvaukseen. Toimintakuvausten oli laatinut Jimexo Tech Oy. Projekti toteutettiin yhteistyössä Jimex Group Oy:n kanssa. Apex Automation Oy:n osuus projektista oli logiikka- ja valvomosuunnittelu sekä raportointi ja tiedonkeruu lastauksesta. Ennen logiikkaohjelman suunnittelua sain lyhyen TIA - Portal koulutuksen Apex Automation Oy:n suunnittelijoilta. Kävimme aluksi läpi Apexin luomia toimilohkoja sekä kuinka WinCC:n face plate objekteja käytetään. Projektin alussa sain myös RFID - koulutuksen. Koulutuksessa kävimme läpi kuinka projektissa käytetyt RFID laitteet toimivat ja kuinka ne otetaan käyttöön. Koulutuksessa käytettiin vielä Siemens Step7 Manageria eikä TIA-Portaalia, joten ohjelmien kääntö TIA:lle tuotti aluksi pieniä vaikeuksia.

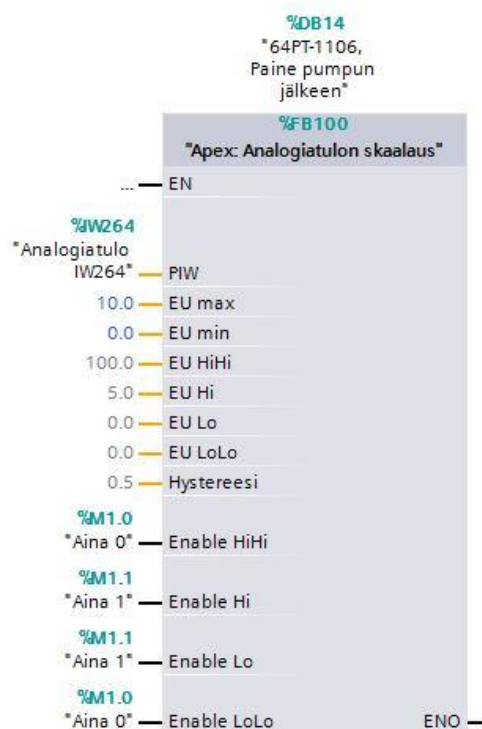
Projektin logiikaksi valittiin aluksi ET200S logiikkaohjain. Meidän täytyi kuitenkin vaihtaa se, sillä Kemira halusi kasvattaa lastausta käyttävien kuskien ja rekka-autojen määrää. Tämä johti siihen, että logiikan muisti tuli täyteen, koska kuljettajat sekä rekka-autojen tiedot ovat logiikan muistissa. Logiikka vaihdettiin ET200S F mallin logiikkaohjaimeen. Sen muistin koko oli suurempi kuin edeltävän logiikan. Kemira halusi kuitenkin edelleen kasvattaa kuljettajien ja rekka-autojen määrää. Seuraava valinta oli Siemens 315-2 PN/DP logiikkaohjain, jonka muistialue oli kaksi kertaa suurempi, kuin aluksi valitun ET200S logiikkaohjaimen. Projektin alussa kuljettajien määrä oli 20 ja rekka-autojen 10. Projektin loputtua kuljettaja paikkoja oli 120 ja rekka-autojen paikkoja 60. Kuljettajien ja rekka-autojen tietojen käsittely logiikassa syö muistia, koska ne ovat string-tyyppisiä muuttujia.

Logiikkaohjelman pohjana käytin Apex Automation Oy:n malliprojektia, koska se sisälsi valmiiksi hyödyllisiä toimilohkoja, joita työssä tarvitsin. Se oli myös jäsennelty järkevästi valmiiksi. Pysin pitämään ohjelman selkolukuisena ja sitä auttoi jo valmis pohja. Ohjelmapohjaan on jaoteltu valmiiksi FC-lohkot, joiden sisälle mikäkin ohjelma tai ohjaus tehdään.

## 7.1 Analogiamittaukset

Analogiamittaukset tulevat logiikalle mA -viestinä, joten pitää ne skaalata oikeisiin arvoihin logiikassa. Käytin mittauksien skaalauksiin Apexin valmista analogiatulon skaalaus FB-lohkoa. Lohko skaalaa siihen kytketyn analogiamittauksen haluttuihin arvoihin. Kuviossa 16 on esillä skaalauslohko painemittaukselle. Lohko skaalaa mittauksen insinööriyksikköihin sekä prosenttiluvuksi 0-100 %. Skaalaus alue määritetään syöttämällä EU max ja EU min tuloihin mittauksen ylä- ja alaraja. Molemmille skaalauksille on omat static-muuttujat, joihin lohko laskee skaalausarvot.

Kyseinen mittaus mittaa pumpun jälkeen olevaa putkiston painetta ja sitä käytetään paineensäätöpiirissä, jota käsitellään pumppujen ohjaukset kappa-leessa. Lastausjärjestelmässä on yhteensä 8 mittausta, jotka kaikki tein samalla periaatteella. Lohkolle voidaan asettaa myös ala- ja ylähälytysrajat. Mittauksien hälytykset tulevat näkyviin valvomonäytön hälytysikkunaan.



KUVIO 16: Analogiatulon skaalauslohko

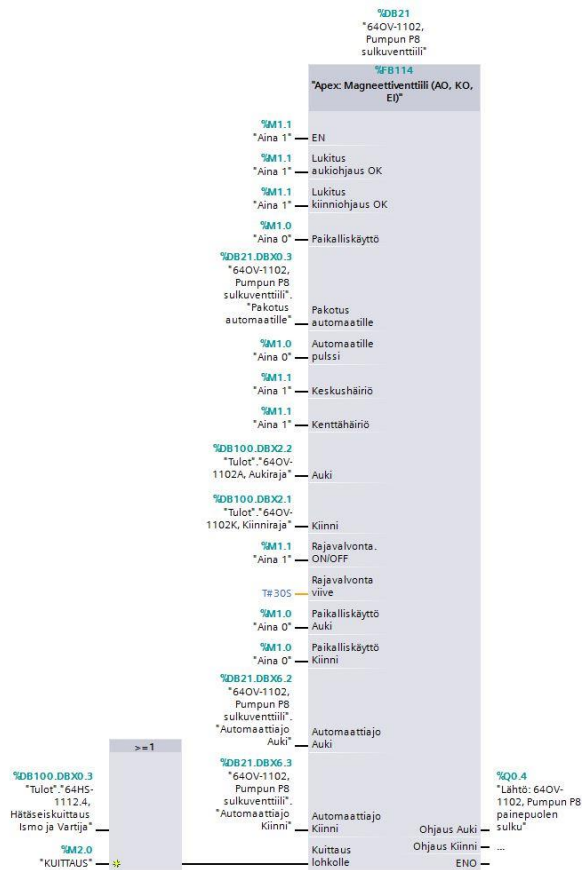
## 7.2 Venttiilien ohjaukset

Lastausjärjestelmä sisältää kahdenlaisia venttiilejä: paineilmalla toimivia on/off-venttiilejä, joissa on jousipalautus sekä paineilmalla toimivan säätöventtiilin, jota ohjataan mA-viestillä. On/off-venttiileihin ohjataan paineilmaa magneettiventtiilillä, jonka kelaan on kytketty logiikan binäärilähtö.

Molemmille ohjaustavoille oli valmiit FB-lohkot, joita hyödynsin ohjelmassa. Venttiilien FB - lohkoilla, kuten kaikilla ohjauslohkoilla on kaksi tilaa; automaatti- ja käsitila. Automaattitilassa venttiiliä ohjataan lohkossa olevien "Automaattiajo Auki" ja "Automaattiajo Kiinni"-nastojen avulla. Käsitilassa venttiiliä ohjataan valvomonäytöltä ohjausikkunan avulla. Automaattiohjauksien



ehdot on rakennettu Automatiikka FC sisälle. Lohkon lähtöpuolen Ohjaus Auki-nastaa kytketään logiikan binäärilähtö, joka on kytkettynä magneetti-venttiiliin, joka ohjaa paineilmaa venttiiliin. Venttiilit sisältävät auki ja kiinni rajatiedot, joilla vaihdetaan, että venttiiliin fyysisesti saavuttaa asennon, johon se on ohjattu. Venttiililohko menee häiriölle jos rajaa ei saavuteta. Häiriötilassa venttiilin auki ohjaus menee pois päältä ja venttiili menee käsitilaan.



KUVIO 17: Pumpun P8 sulkuventtiilin ohjauslohko

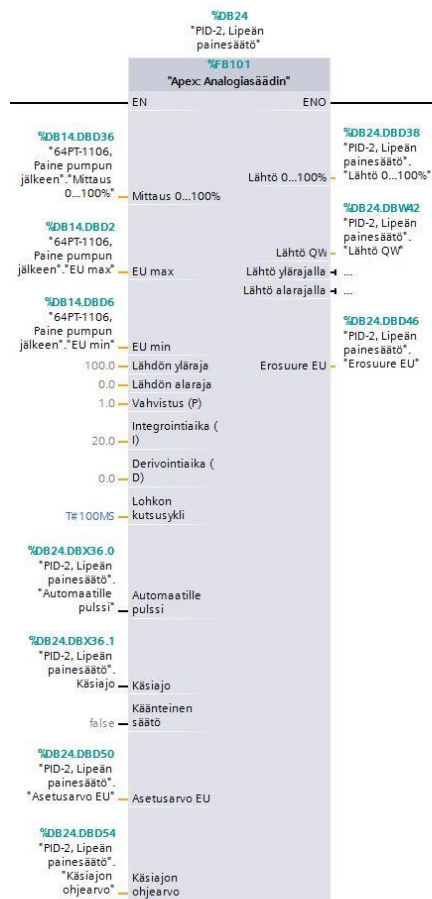
Lastauksessa olevalla säätöventtiilillä piti alun perin säätää lastauksen virtausnopeutta. En saanut kuitenkaan säädintä toimimaan yhdessä paineensäädön kanssa, joten säätöpiiristä luovuttiin. Kun lastaus käynnistetään, säätöventtiili avataan 15 sekunnin aikana auki asentoon askeleittain. Säätöventtiili suljetaan myös askeleittain, kun lastattu määrä lähenee asetettua lastaus mää-

rää. Näin saadaan hallitusti pysäytettyä lastaus eikä putkistoon kohdistu paineiskua.

### **7.3 Pumpun ohjaukset**

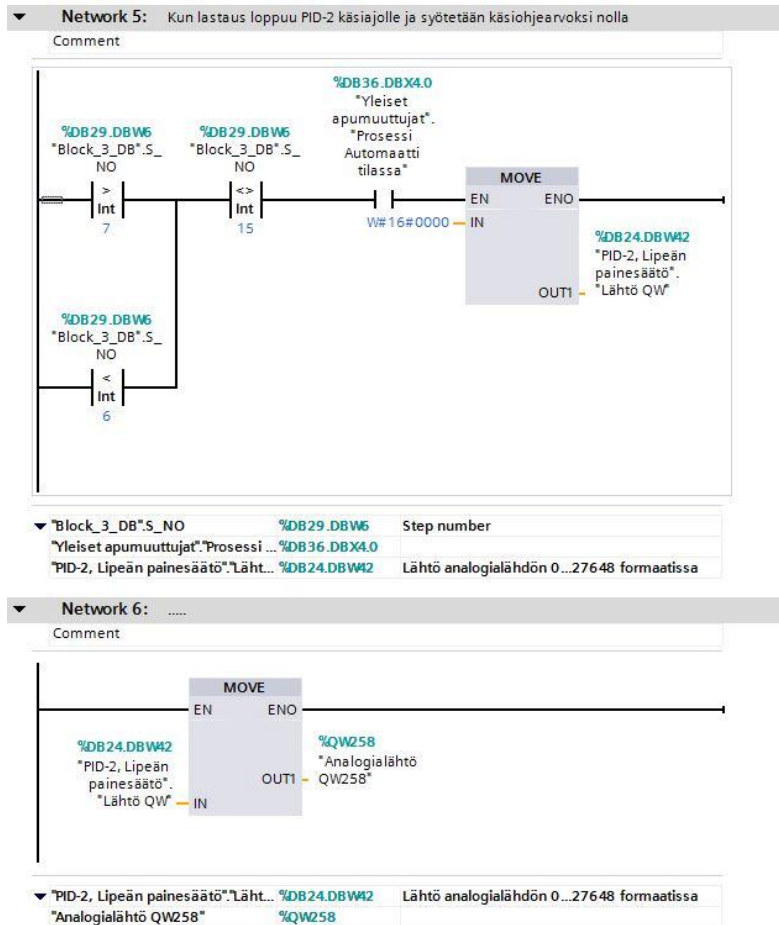
Lastausjärjestelmä sisältää kaksi pumppua, joista vain toinen on kerrallaan käytössä. Pumpuilla on yhteinen taajuusmuuttaja, joten ohjelmaan tarvitsi tehdä vain yhden pumpun ohjaukset. Pumpun valinta suoritetaan paikalliskytkimellä. Lastauspumppua ohjataan käyntiin antamalla taajuusmuuttajalle ohjauskäsky sekä haluttu nopeusohje.

Lastauspumppua ohjataan paineensäätöpiirillä, jolla säädetään lastauspainetta. Säättöpiirille annetaan paineen asetusarvo, jota säättöpiiri pyrkii lastauspumpon kierrosnopeutta säätämällä pitämään. Apexilta löytyi oma analogiasäädin FB – lohko, jota käytin työssäni (Kuvio 18).



KUVIO 18: Analogiasäädin

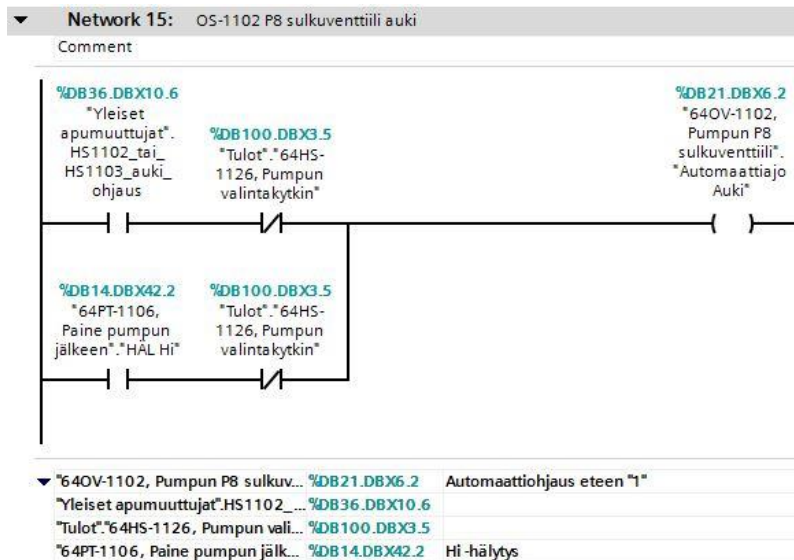
Säädinlohkolle syötetään säädettävämittausta, joka on kyseissä säätöpiirissä paine pumpun jälkeen. Säätöpiirin vahvistus, integrointi- sekä derivointiaika syötetään valvomonäytöltä säätimen ohjausikkunasta. Säätimen "Lähtö QW" -nastan syötetään logiikan analogialähtöosoite, joka lähettää taajuusmuuttajalle mA-viestinä asetusarvon pumpun kierrosnopeudelle. En kuitenkaan suoraan kytkenyt lähtöosoitetta säädinlohkon "Lähtö QW"-nastan. Tein varmistuksen, että vain sekvenssin askeleissa, kun lastaus on päällä, pumpulle annetaan kierrosnopeus asetus säätimeltä. Muissa sekvenssin askeleissa lähtöosoitteeseen syötetään nolla (Kuvio 19).



KUVIO 19: Pumpun kierrosnopeuden syötönesto

## 7.4 Automaattiohjaukset

Lastauksessa tapahtuvat automaattiset ohjaukset tein suurimmaksi osaksi Automaatiikka FC:n sisälle. Ohjauksille rakennetaan ehdot, joiden täyttyessä ohjaus toteutetaan. Kuviossa 20 on esillä lastauspumppu P8 sulkuventtiilin automaattiajo auki. Venttiilille annetaan auki ohjaus - käsky, kun sekvenssissä ohjattava (Yleiset apumuuttujat.HS1102\_tai\_HS1103\_auki\_ohjaus) apumuuttaja tai paine pumpun jälkeen hälytys (64PT-1106, Paine pumpun jälkeen.Häl Hi) on päällä sekä pumpun valintakytkin (Tulot.64HS-1126) on pois päältä.



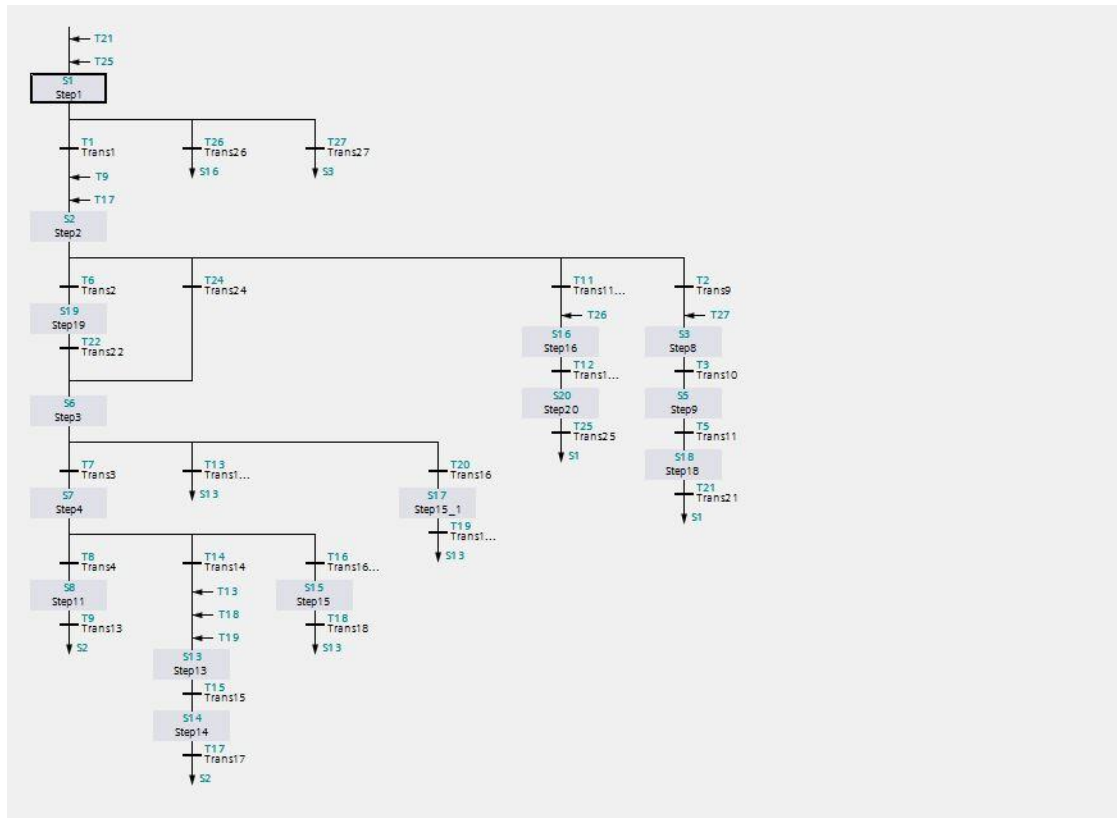
KUVIO 20: P8 sulkuventtiilin automaattiajo Auki

## 7.5 Sekvenssi

Sekvenssi koostuu askeleista, joista vain yksi kerrallaan on aktiivinen. Askeleet taas sisältävät toimenpiteitä, joita suoritetaan askeleen aikana. Askeleissa edetään eteenpäin siirtymäehtojen täytyttyessä. Sekvenssi ohjelmointia varten Siemensillä on GRAPH – työkalu.

Sekvenssin askeleissa ohjataan apumuuttujia, jotka ovat ohjausehtoina Automaattikka FC:n sisällä olevissa venttiilien ja pumppujen ohjauksissa.

Kun lastaus käynnistetään, askelletaan sekvenssissä askeleeseen jossa käynnistetään pumppu sekä avataan venttiilit. Askeleessa odotetaan siihen asti, kunnes venttiilit ovat auenneet. Tämän jälkeen askelletaan askeleeseen jossa odotetaan että haluttu lastaus määrä on saavutettu. Kun lastaus määrä on saavutettu, suljetaan seuraavissa askeleissa venttiilit ja pumppu. Kun venttiilit ovat sulkeutuneet, askelletaan sekvenssin alkuun. Lastauksen sekvenssikaavio on esitetty kuviossa 21.



KUVIO 21: Sekvenssikaavio

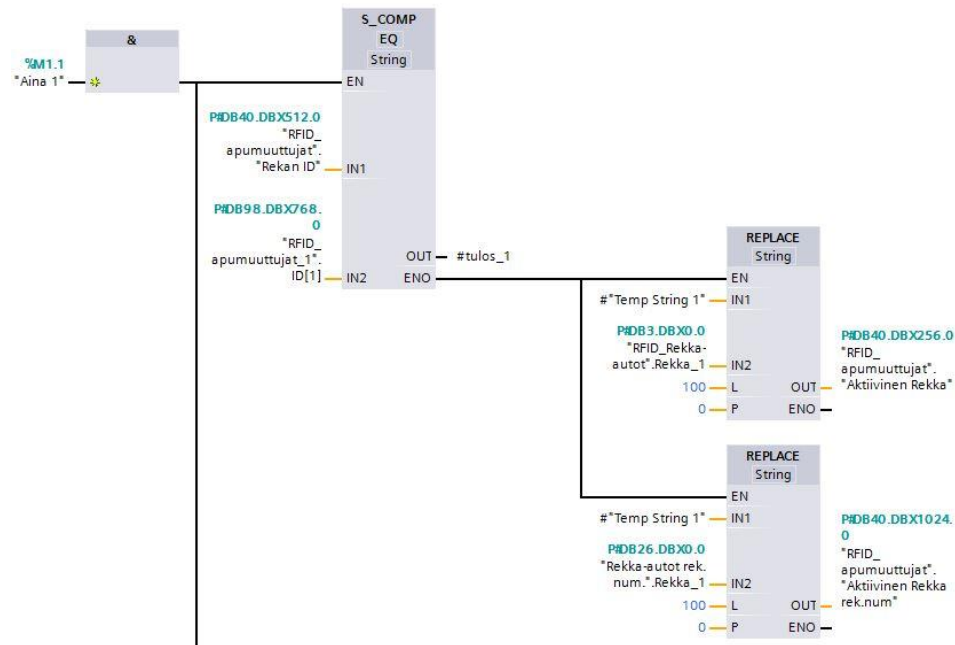
## 7.6 RFID käsittelyt

Kun RFID-lukija saa luettua tunnistuksessa olevan ID-numeron, haetaan sen perusteella ID-numeroon sidotut tiedot. Henkilötunnisteet koostuvat luvuista 1-120 ja rekka-autojen tunnistukset 24 merkin string-tyyppisestä merkkijonosta. Kuljettajien nimille, rekka-autojen kuljetusliikkeille sekä rekisterinumeroille on omat datablokkinsa, joihin tiedot on koottu. Datablokkien sisältöä voidaan muuttaa valvomonäytöltä käsin. Tiedot haetaan vertailemalla luettua ID-numeroa siihen määrään lukuja, kuin on tunnistetta käytössä.

Henkilötunnisteen ID-numero tallennetaan Data.Daten[1]-muuttuun, joka on data tyyppiltään Byte. Sen formaatti on hexa-muodossa. Muuttujaa verra-



S\_COMP on string data tyyppien vertailulohko ja se ei salli, että IN-nastoihin kirjoittaa suoraan vertailtavan arvon, vaan siihen täytyy tuoda muuttuja. Tämän takia jouduin luomaan ylimääräisen RFID\_apumuuttuja\_1 datablokin, joka sisältää vastaavat string-tyyppiset muuttujat kuin mitä rekka-autojen tunnisteisiin on kirjoitettu.



KUVIO 22: Rekka – auton tunnisteiden vertailu

Kuviossa 22 on esillä rekka-autojen ID-numeroiden vertailu. Jos luetun tunnisteiden ID-numero on 000000000000000000000001, siirretään kuljetusliikkeen tiedot sisältämästä datablokista (RFID\_Rekka-autot) paikalla 1 olevan muuttujan tiedot muuttujaan, joka ilmoittaa järjestelmään kirjautuneen rekka-auton (RFID\_apumuuttujat.Aktiivinen Rekka). Samalla siirretään myös auton rekisterinumero järjestelmään.



## 7.7 Valvomosuunnittelu

Valvomosuunnittelua varten TIA-Portaali sisältää WinCC valvomo-ohjelmiston, jolla suunnitellaan valvomokäyttöliittymä. WinCC soveltuu kaikille teollisuussektoreille ja teknologialle. Se mahdollistaa sekä vaativat yksittäissovellukset, että monimutkaiset usean operaattorin järjestelmät (Simatic WinCC, Siemens Oy:n nettisivut).

Valvomo on projektissa suhteellisen yksinkertainen. Se sisältää prosessikaavio-, parametrit -, hälytykset-, taagit -, ja kirjautuminen - sivut. Parametrit - ja taagit - valvomonäytöt on suojattu salasanalla. Sivunvaihto tapahtuu yläreunassa olevilla painikkeilla.

WinCC sisältää myös mahdollisuuden käyttää VB scriptejä. VB Script on komentosarjojen suorittamisessa käytettävä ohjelmointikieli ja se on kevennetty versio Microsoftin Visual Basic ohjelmointikielestä. (Johdatus VBScriptiin, 2kmediat kotisivu). Scriptien avulla luotiin toiminnot rahtikirjan tulostukseen, lastaustietojen kirjoitukseen tietokantaan sekä asiakastietojen tuonti tietokannasta logiikkaan.

### 7.7.1 Prosessikaavio-sivu

Valvomosuunnittelun aloitin prosessikaavio - sivun suunnittelulla. Kyseinen näyttö on valvomon pääsivu. Näytön pohjakuvassa hyödynsin projektin PI-kaaviota. PI-kaaviossa kuvataan prosessin putkisto ja instrumentointi. Prosessikaavio - sivulla on näkyvillä kaikki prosessin venttiilit, pumput ja mittapisteet. Lastausta suorittavaa kuljettajaa varten valvomonäytölle piti lisätä I/O-

kentät, josta nähdään kuljettajan nimi ja kuljetusliike, jotta kuljettaja voi tarkistaa tiedot oikeiksi.

Valvomonäytöllä oleville toimilaitteille ja mittauksille tein ohjausikkunat eli face platet. Ohjausikkunan avulla voidaan esimerkiksi venttiili asettaa tarvittaessa käsitilaan, jolloin sitä voidaan ohjata kiinni tai auki valvomosta. Mittauksille voidaan ohjausikkunan avulla asettaa hälytysrajat.



KUVIO 23: Venttiilin ohjausikkuna

### 7.7.2 Taagit-sivu

Tunnisteiden hallinta tapahtuu Taagit sivulta. Sivulla voidaan muuttaa tunnisteesiin sidottujen henkilöiden ja rekka-autojen tietoja sekä voidaan muuttaa tunnisteen id - numeroita. Ensimmäiseksi sivulle avautuu henkilötunnisteen hallintasivu. Sivut on estetty salasanalla, ettei kuka tahansa henkilö pääse muuttamaan tunnisteesiin liitettyjä tietoja. Sivulla on lueteltuna kaikki käytössä olevat RFID - tunnistet ja niiden edessä tunnisteeseen liitetty kuljettaja. "Rekka-autot"-painikkeella päästään rekka-autojen tietoihin käsiksi. Ohjeet tunnisteen hallintaan löytyy liitteestä 4.

### 7.7.3 Hälytykset-sivu

Hälytykset-sivulla on näkyvillä aktiiviset hälytykset prosessissa. Sivulta nähdään samat hälytykset kuin prosessikaavio-sivun alalaidassa olevasta hälytysikkunasta. Tärkeimmät hälytykset lähetetään eteenpäin lastausalueen valvojalle tai vartijalle Apex SMS järjestelmän avulla.

### 7.7.4 Parametrit-sivu

Parametrit-sivulta voidaan asettaa lastauspaineelle asetusarvo.

### 7.7.5 Scriptit

Lastauksen tiedot raportoidaan kolmeen paikkaan logiikalta: Rahtikirjaan, MySQL-tietokantaan sekä tekstitiedostoon. Minun työkseni jäi rahtikirjan ja MySQL-tietokantaan kirjoittaminen.

Kemira halusi päästä näkemään nettiselaimen kautta toteutuneita lastauksia. Tämän johdosta luotiin MySQL-tietokantaan LastausPaivakirja taulu, johon kirjataan jokaisesta lastauksesta tiedot, joita Kemira Oy:n väki halusi nähtäväksi. Tietokannan tietojen siirron nettiselaimen ei kuulunut minun työhöni.

Lastauksen tapahtuman tiedot MySQL-tietokantaan kirjoitetaan WinCC:ssä käytössä olevilla VB scripteillä. Opastusta tietokantaan kirjoituksesta sain Apex Automation Oy:n suunnittelijoilta.

Tietokantaan kirjoitetaan lastausaika, lastattu määrä, tositenumero, rekisterinumero, kuljetusliike (rekka-auton tunnisteen ID-numero), asiakas, paikkakunta sekä lastauskohtainen numero.

Scriptin alussa näille tiedolle luodaan scriptissä muuttuja (Liite 2, rivit 18). Näihin muuttujiin liitetään valvomomuuttuja, josta luetaan tiedot muuttujiin (Liite 2, rivit 27-48). Tämän jälkeen luodaan yhteys tietokantaan ja kirjoitetaan muuttujat LastausPaivakirja nimiseen tauluun tietokannassa (Liite 2, rivit 50-60). Lopuksi scriptissä vielä suljetaan yhteys tietokantaan.

Scriptin suoritetaan valvomossa, kun lastaus päätetään ja rahtikirjat tulostetaan.

### 7.7.6 Rahtikirjan tulostus

Wincc sisältää raportin suunnittelutyökalun, jonka avulla toteutettiin rahtikirjan tulostus. Sain Kemira Oy:ltä heidän käyttämänsä rahtikirjamallin, jonka perusteella loin raportille pohjan. Pohja sisältää tekstikenttiä sekä I/O-kenttiä. Tekstikentät ovat vakiotekstejä, jotka tulostuvat jokaiselle rahtikirjalle. I/O-kentät on yhdistetty valvomomuuttujiin, joihin tieto haetaan logiikan muistista. Rahtikirjan tulostus suoritetaan VB scriptin avulla. Jokaisesta lastauksesta tulostetaan kolme rahtikirjaa. (Liite3)

## 8 IO-TESTAUS

Lastausaseman IO-testaus aloitettiin, kun aseman sähköiset kytkennät olivat valmiina. IO-testauksessa käytiin läpi, että logiikkaan kytketyt anturit ja mitaukset sekä logiikan lähtöihin kytketyt ohjaukset olivat kytketty oikein. Suoritin I/O-testauksen yhdessä Jimexon Group:in suunnittelijan kanssa. Tulostin I/O-testausta varten I/O - listan, jossa oli lueteltuna kaikki logiikkaan kytketyt I/O-pisteet ja kävimme listan läpi järjestyksessä. Työnjakomme oli, että Jimexon Group:in suunnittelija oli kentällä ja kytki antureita päälle sekä seurasi venttiilien liikkeitä ja minä tarkistin logiikalta tuliko anturia vastaava tulo

päälle sekä ohjailin venttiilejä valvomosta. I/O-testaukseen oli varattuna pari päivää aikaa ja se riitti meille hyvin.

RFID-laitteita testatessamme huomasimme, ettei rekka-auton tunnistukseen tarkoitettu lukija saanut luettua tunnistetta. Soitin asiasta Siemensille ja sieltä neuvottiin tarkistamaan antennien lähetysteho, jonka olin asettanut liian pieni. Kun tuplasin lukijan käyttämän lähetystehon, rupesi lukija toimimaan normaalisti.

I/O-testauksen päätteeksi asetimme RFID-tunnisteille valmiiksi ID-numerot sekä lisäilimme tunnisteisiin liitettävien kuljettajien ja kuljetusliikkeiden nimiä järjestelmään.

## 9 KÄYTTÖÖNOTTO

Lastauksen käyttöönotto aloitettiin noin pari päivää I/O-testauksen jälkeen. Käyttöönotossa oli tarkoitus testata järjestelmän toimivuus sekä opettaa lastausasemanhoitaja sekä kuljettajan käyttämään lastausjärjestelmää.

Käyttöönotto aloitettiin käsiajojen testauksella, että saatiin ilma pois lastausputkistosta. Avasimme kaikki venttiilit auki asentoon ja annoimme lipeän valua omalla painollaan lastattavaan rekka-autoon. Samalla opetin lastausasemanhoitajalle kuinka toimilaitteet asetetaan käsiajotilaan.

Ensimmäisen automaattiajon aikana huomasimme, ettemme saa molempia säätöpiirejä toimimaan yhdessä. Luovuimme virtaussäätöpiiristä ja muutimme säätöventtiilin ohjausta siten, että se avautuu auki asentoon askeleittain lastauksen käynnistyttyä. Säätöventtiilin kiinniajo on sidottu jäljellä olevaan lastattavan tavarán määrään. Kun jäljellä on enää tuhat kiloa lastattavaa lipeää, aletaan säätöventtiiliä askeleittain ajaa kiinni.

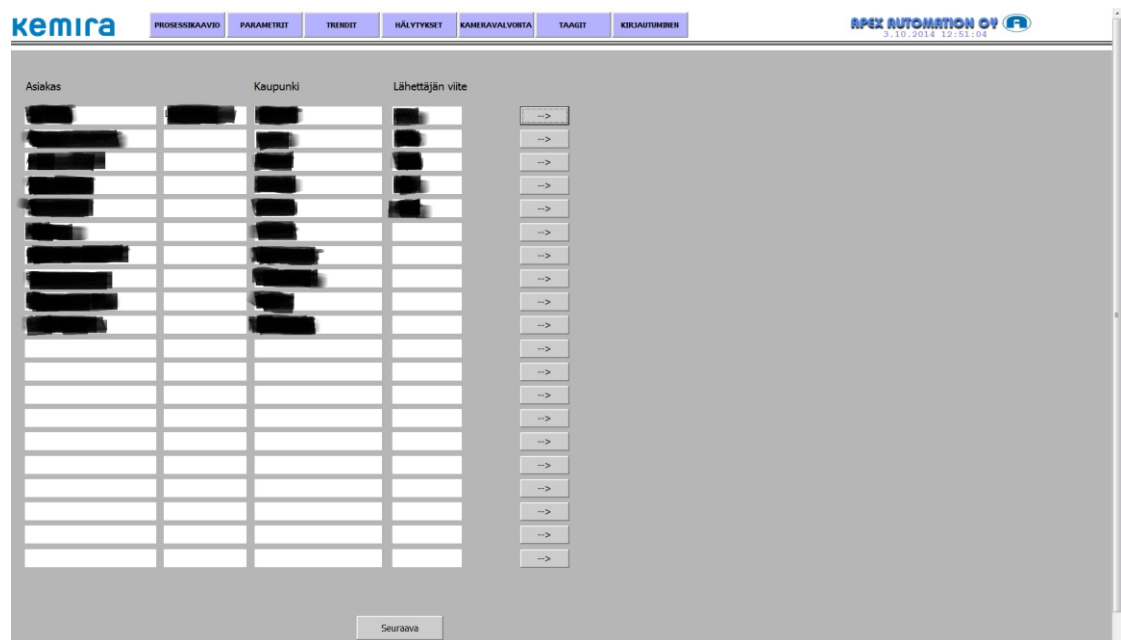
Toinen ongelma, joka tuli vastaan oli ulkokaivon lipeänmittaus. Ulkokaivossa oleva mittaus mittaa veden lipeänpitoisuutta, joka valuu kaivoon, kun kuljettaja puhdistavat vedellä autojaan lastauksen päätteeksi. Jos mittauksen lipeäpitoisuus on liian suuri, lastausta ei voida käynnistää. Tämä todettiin turhaksi lukitukseksi, joten poistimme sen lastauksen käynnistysehdoista.

Samalla kun lastausasemalla kävi rekka-autoja, annoimme heille autoihin tunnisteet sekä henkilökohtaisen tunnisteiden sekä opetimme heille kuinka lastausasemaa käytetään.



Kun henkilötunnistus on tehty, ilmestyy näytölle ”Tilaukset”-painike (Kuvio 24), josta kuski pääsee valitsemaan asiakkaan kenelle lastattu tavara toimitetaan.

Kuljettaja etsii asiakkaan listalta ja valitsee sen painamalla asiakkaan oikeassa reunassa olevaa nuolipainiketta (Kuvio 25).



KUVIO 25. Valvomonäyttö, asiakaslista

Asiakkaan valinnan jälkeen tulee näkymä rahtikirjaan tulostettavista tiedoista. Kuljettaja voi vielä tässä vaiheessa vaihtaa asiakasta tai muuttaa tekstikentissä olevia tietoja. Kuski valitsee asiakkaan klikkaamalla ”Valitse asiakas” - painiketta (KUVIO 26).



KUVIO 26. Valvomonäyttö, rahtikirjan pohja

Rahtikirjan valinnan jälkeen kuljettaja asettaa lastattavan määrän ja aloittaa lastauksen klikkaamalla "Käynnistä lastaus" - painiketta. Kun lastaus käynnistyy, avataan lipeäsäiliön juuriventtiili ja pumpun sulkuventtiili. Lastauspumppu käynnistetään sekä säätöventtiili avataan ensiksi 30 prosenttiin. Pumpun painesäädölle annetaan asetusarvoksi 1,3 bar:ia. Säätöventtiili avataan täysin auki 15 sekunnin aikana lastauksen käynnistyksestä. Kun säätöventtiili on täysin auki, logiikka antaa pumpun painesäädölle asetusarvoksi 1,6 bar:ia. Lastaus käynnistetään hiljalleen, ettei äkillinen paine nosta lastausvartta pois rekka-auton säiliöstä.

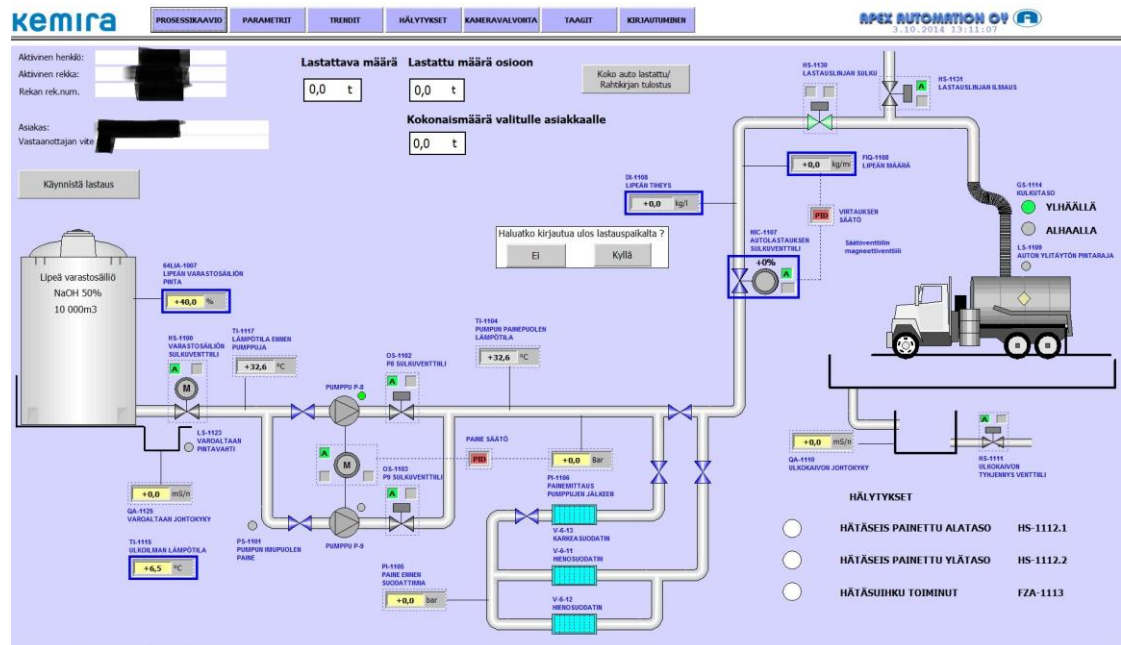
Kun lastaus on käynnistetty, on kuljettajan painettava KM-painiketta (kuolleen miehen painike) 60 sekunnin välein. Tällä varmistetaan, että lastaus pysähtyy, jos kuljettaja esimerkiksi nukahtaa tai satuttaa itseään lastauksen aikana. Jos painiketta ei paineta, lastaus keskeytetään sulkemalla säätöventtiili portaittain sekä lastauspumppun kierroslukua pienentämällä portaittain alas-

päin. Kun säätöventtiili on päässyt kiinni asentoon, suljetaan pumpun sulkuventtiilin sekä lastauspumppu pysäytetään. Jos venttiilit suljettaisiin saman tien hälytyksen tultua, rasittaisi se putkistoa liikaa paineiskun johdosta.

Lastatun lipeän määrää mittaa Ontec Oy:n tekemä mittausjärjestelmä. Lastauksen sisältämä massavirtamittari on kytketty heidän logiikkaansa, josta he lähettävät lastatun lipeän määrää meidän käytössä olevaan logiikkaan. Tähän järjestelyyn päädyttiin, koska lastaukset menevät kaupalliseen käyttöön ja siihen tarvitaan sertifioitu mittausjärjestelmä, joka oli tarjolla Ontec Oy:llä.

Kun lastattu määrä on enää tuhat kiloa halutusta määrästä, aletaan lastausta hidastaa. Hidastus tapahtuu sulkemalla säätöventtiiliä askeleittain lastausmäärän lähestyessä haluttua määrää. Kun haluttu lipeän määrä on saavutettu, suljetaan pumpun sulkuventtiili, lastauspumppu sekä säätöventtiili.

Kun lastaus on päättynyt, kuljettajalla on kolme eri vaihtoehtoa. Hän voi aloittaa uuden lastauksen asettamalla uuden lastattavan määrän ja käynnistämällä lastauksen, tulostaa rahtikirjat ja jatkaa lastausta uudelle asiakkaalle tai tulostaa rahtikirjat ja kirjautua ulos lastausjärjestelmästä. Jos kuljettaja haluaa jatkaa lastausta toiselle asiakkaalle, kuljettaja painaa "Koko auto lastattu/ Rahtikirjan tulostus"-painiketta ja vastaa "Haluatko kirjautua ulos lastauspaikalta?"-kysymykseen "Ei" (Kuvio 20). Tällöin asiakastiedot sekä lastattu määrä tiedot nollataan näytöltä ja edellisen lastauksen rahtikirjat tulostuvat. "Tilaukset"-painike ilmestyy näytölle ja kuljettaja valitsee asiakkaan uudelleen. Jos kuljettaja haluaa kirjautua ulos lastauspaikalta, vastaa hän "Haluatko kirjautua ulos lastauspaikalta?"-kysymykseen "Kyllä" (Kuvio 20). Tällöin asiakastiedot, lastausta käyttävän kuljettajan henkilö ja rekka-auton tiedot sekä lastattu määrä häviävät näytöltä ja rahtikirjat tulostuvat.



KUVIO 27: Valvomönäyttö, uloskirjautuminen lastauspaikalta

## 11 POHDINTA

Projekti soveltui opinnäytetyöksi erittäin hyvin. Työ oli erittäin laaja kattaen logiikka- ja valvontasuunnittelun ja RFID:n soveltamisen lisäksi MySQL-tietokantaan kirjoittamisen Wincc VB scripteillä. Työn aikana sain hyvän käsityksen siitä millainen on automaatio suunnittelijan toimenkuva.

Projekti eli matkan varrella erittäin paljon. Alun perin työ sisälsi ainoastaan 20 kuljettajan tunnisteiden sekä 10 rekka-auton tunnisteiden käsittelyn. Nämä luevat kuitenkin kasvoivat työn edessä reilusti. Tämä aiheutti hieman päänvaivaa, sillä logiikan muisti täyttyi tunnisteiden kasvaessa ja logiikkaa täytyi vaihtaa muutamaaan otteeseen. Jälkeenpäin mietittynä olisi ollut järkevämpää ylläpitää MySQL-tietokannassa kuljettajien ja rekka-autojen tietoja. Tällöin olisi välttytty turhilta logiikan vaihdoilta.

Projektiin ei alun perin sisältynyt lastaustapahtumien talteenottoa tietokantaan eikä asiakastietojen ylläpitoa tietokannassa vaan nämäkin tulivat ns. lisätöitä projektin edessä. Aluksi oli suunniteltu, että kuljettaja kirjoittaisi rahtikirjaan asiakkaan tiedot käsin. Tämä oli kuitenkin mukava lisäys, sillä pääsin toteuttamaan tietokantaan kirjoittamista, jota en aiemmin ole tehnyt.

RFID-laitteiden käyttöönotto projektissa sujui hyvin. Aluksi oli ongelmia RF670R lukijan lukuetaisyyden kanssa, koska olin parametroinut antennille liian pienen lähetystehon. Aikaa olisi saanut olla hieman enemmän, jotta olisi paremmin päässyt tutustumaan laitteisiin. Lukijat ovat kuitenkin toimineet paikan päällä moitteettomasti joten voin olla tyytyväinen. Palautetta on hieman tullut lastausalueen hoitajalta koskien tunnisteiden hallinta ohjeita.

## LÄHTEET

Apex Automation Oy kotisivut. Viitattu 8.4.2013. [www.apexautomation.fi/fi](http://www.apexautomation.fi/fi)

Konserni. Kemiran kotisivut. Viitattu 8.4.2013

<http://www.kemira.com/fi/konserni/Sivut/default.aspx> (Viitattu 8.4.2013)

RFID – tietoutta. RFIDLab kotisivut. Viitattu 18.4.2013.

<http://www.rfidlab.fi/rfid-tietoutta>

RFID – tekniikan historia, RFIDLab kotisivut. Viitattu 18.4.2013.

<http://www.rfidlab.fi/rfid-tekniikan-historia>

RFID – tekniikan fysikaalliset perusteet, RFIDLab kotisivut. Viitattu 18.4.2013.

<http://www.rfidlab.fi/rfid-tekniikan-perusteet>

RFID – tekniikan käyttämät taajuusalueet, RFIDLab kotisivut. Viitattu 18.4.2013.

[http://www.rfidlab.fi/rfid-tekniikan-käyttämätt-taajuusalueet](http://www.rfidlab.fi/rfid-tekniikan-kayttamatt-taajuusalueet)

RFID-etätunnistusteknologia. Tekniikan maailma 6.2012, s.110 - 114

Johdatus VBScriptiin, 2kmediat kotisivut. Viitattu 5.11.2014.

<http://www.2kmediat.com/vbscript/johdanto.asp>

Simatic WinCC. Siemens Oy. Viitattu 5.11.2014.

[http://www.siemens.fi/fi/industry/teollisuuden\\_tuotteet\\_ja\\_ratkaisut/tuotesivut/automaatiotekniikka/kayttoliittymat/ohjelmistot/valvomo\\_ohjelmisto\\_winc.c.php](http://www.siemens.fi/fi/industry/teollisuuden_tuotteet_ja_ratkaisut/tuotesivut/automaatiotekniikka/kayttoliittymat/ohjelmistot/valvomo_ohjelmisto_winc.c.php)

HOME ,PLCTutor. Viitattu 7.11.2014. <http://www.plctutor.com/>

TIA-Portal Suunnitelun uusi aikakausi, Viitattu 10.4.2013.

[http://www.siemens.fi/pool/products/industry/iadt\\_is/tuotteet/tia\\_portal/tia\\_portal\\_esite.pdf](http://www.siemens.fi/pool/products/industry/iadt_is/tuotteet/tia_portal/tia_portal_esite.pdf)

Keinänen, T., Kärkkäinen, P., Lähetkangas, M. & Sumujärvi, M. 2007. Auto-maatiojärjestelmien logiikat ja ohjaustekniikat. Helsinki: WSOY

Helipotusta sosiaali ja terveystalveluiden arkeen rfidn avulla, RFIDLab kotisivut. Viitattu 2.12.2014.

<http://www.rfidlab.fi/blogi/helipotusta-sosiaali-ja-terveystalveluiden-arkeen-rfidn-avulla>

RFID. MikroVäylän kotisivu. Viitattu 2.12.2014.

<http://www.mikrovayla.fi/kirjastot/teknologiat.php>

Tehokkuutta toimitusketjun hallintaan ja kaupankäyntiin RFID – tekniikalla, Jukka Wallinheimo. Viitattu 2.12.2014.

[http://logistiikkamessut.fi/liitetiedostot/editori\\_materiaali/1779.pdf](http://logistiikkamessut.fi/liitetiedostot/editori_materiaali/1779.pdf)

Programming Guideline 2014. Programming for S7-1200/1500. Viitattu 20.10.2014.

[http://cache.automation.siemens.com/dnl/zE/zExMzE4MQAA\\_81318674\\_Tools/81318674\\_Programming\\_guideline\\_DOKU\\_v12\\_en.pdf](http://cache.automation.siemens.com/dnl/zE/zExMzE4MQAA_81318674_Tools/81318674_Programming_guideline_DOKU_v12_en.pdf)

Siemens Support. Tuotokuva Siemensin internetsivuilta. Viitattu 2.7.2014.

<http://support.automation.siemens.com/WW/llisapi.dll?func=cslib.csinfo&objId=36816516&nodeid0=33516848&load=content&lang=en&siteid=cseus&aktprim=0&objaction=csview&extranet=standard&viewreg=WW>

Siemens tuoteuutinen, 2010. Tuoteuutinen Siemensin internetsivuilla. Viitattu 2.7.2014.

[http://www.siemens.fi/fi/industry/teollisuus/tuoteuutiset/erittain\\_suosittu\\_et2005s\\_cpu\\_paivitty.htm](http://www.siemens.fi/fi/industry/teollisuus/tuoteuutiset/erittain_suosittu_et2005s_cpu_paivitty.htm)

Aksprom. Tuotokuva akspromin internetsivuilta. Viitattu 2.7.2014.

<http://www.aksprom.biz/Catalog/Siemens/16459/SIMATIC-RF180C>

Siemens Support, 2012. Tuotokuva Siemensin Supportin internetsivuilta. Viitattu 2.7.2014.

<http://support.automation.siemens.com/US/lisapi.dll?func=cslib.csinfo&objId=60464282&nodeid0=43532183&load=content&subtype=133400&lang=en&siteid=cseus&aktprim=0&objaction=csview&extranet=standard&viewreg=US>

Siemens Parts Online. Tuotokuva Siemens Parts Online internetsivuilta. Viitattu 8.7.2014. <http://www.siemenspartsonline.com/product/6gt2811-0ab00-0aa0-siemens/>

TPAutomation. Tuotokuva TPAutomationin internetsivuilta. Viitattu

25.8.2014. [http://www.tpautomation.de/shop/RFID-Systeme/SIMATIC-](http://www.tpautomation.de/shop/RFID-Systeme/SIMATIC-RF600/RF600-Antennen/6GT2812-0AA00-SIMATIC-RF600-Antenne-RF660A-865-868-EU-o-Kabel::15126.html)

[RF600/RF600-Antennen/6GT2812-0AA00-SIMATIC-RF600-Antenne-RF660A-865-868-EU-o-Kabel::15126.html](http://www.tpautomation.de/shop/RFID-Systeme/SIMATIC-RF600/RF600-Antennen/6GT2812-0AA00-SIMATIC-RF600-Antenne-RF660A-865-868-EU-o-Kabel::15126.html)

Siemens Support. Tuotokuva Siemensin internetsivuilta. Viitattu 2.9.2014.

<http://support.automation.siemens.com/US/lisapi.dll?func=cslib.csinfo&objId=18991929&nodeid0=10805827&load=content&lang=en&siteid=cseus&aktprim=0&objaction=csview&extranet=standard&viewreg=US>

Siemens Industry mall. Tuotekuva Siemensin internetsivuilta. Viitattu  
2.9.2014.

<https://mall.industry.siemens.com/mall/en/WW/Catalog/Product/6GT2810-2HC81>



# LIITTEET

## Liite 1. Ohje RF670R konfigurointi

Ensimmäiseksi ohjelmassa luodaan projekti. Projektin luontisivu aukeaa ensimmäiseksi, kun ohjelma avataan. Projektille valitaan konfiguroitavan lukijan tyyppi, sekä radio profiili joka on maakohtainen. Tässä projektissa lukijaksi valitaan RF670R ja radio profiilin maaksi valitaan Suomi ja radio profiiliksi ETSI.

Kun projekti on luotu, päästään asettamaan lukijalle parametreja. Ensimmäiseksi annetaan lukijalle ip – osoite. Osoite saadaan asetettua yläpalkin Readers valikon alta löytyvästä Configure Reader Address kohdasta. Laite täyttyy aluksi etsiä Start Discovery painikkeella. Kun laite löytyy, se valitaan Use

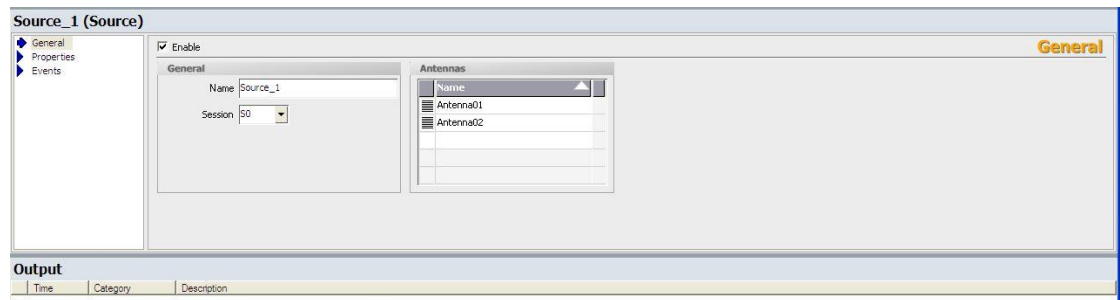
Reader Address painikkeella. Tämän jälkeen lukijalle annettava ip – osoite asetetaan IP Settingsin kenttiin ja painetaan Set Reader IP Address painiketta.

IP- osoitteeksi annetaan osoite, joka on samassa osoiteavaruudessa logiikan kanssa. Kun osoite on asetettu, voidaan varmistaa saiko lukija annetun osoitteen Start Discovery painikkeen avulla.

Kun osoite on annettu, parametroidaan käytössä olevat antennit lukijalle. Antennas välilehdestä pystytään hallitsemaan antennien asetuksia. Antennille asetetaan säteilyteho (erp), vahvistus, RSSI kynnysarvo ja kaapelihäviö. Simatic RF600 manuaalista löytyy sopivat asetukset antennille.

RF-MANAGER Basic V2 - Project.rfidb				
Project Edit Reader View Options Help				
<div> <div>RFID device</div> <div>Antennas</div> <div>Sources</div> <div>I/O Ports</div> <div>Notification channels</div> <div>Trigger</div> <div>Data Selector</div> <div>Tag Selector</div> </div>				
Name	Power level	Gain (dBi)	Cable loss (dB)	Description
Antenna01	1300 mW	7.0	2.0	Antenna 1
Antenna02	1300 mW	7.0	2.0	Antenna 2
Antenna03	100 mW	7.0	2.0	Antenna 3
Antenna04	100 mW	7.0	2.0	Antenna 4

Seuraavaksi valitaan Sources välilehti. Välilehdessä Sourcelle annetaan nimi ja valitaan mitä antennoja lukija käyttää.



Parametrit ladataan lukijalle ylävalikon nuolipainikkeesta.



Lukijaa pääsee testaamaan ylävalikon silmälasipainikkeesta.



## Liite 2. Lastauspäiväkirja scriptti

```

13 Dim objConnection 'yhteys objekti
14 Dim objRecordset 'kyselyn tulos joukko
15 Dim objCommand 'komento objekti
16 Dim yhteys 'yhteys määrittäminen
17 Dim lisays 'sql lisäys
18 Dim maara, ID, Aikaa, Lastausmaara, Tositenumero, RekNum, Kuljetusliike, Asiakas, Paikkakunta
19 Dim aika
20 Dim Numero
21 Dim Kello
22 Declares variables and allocates storage space.
23 Dim LogattuAikal
24 Dim Aika2, Aika3, Aika4, Aika5, Aika6, Aika7
25
26
27 LogattuAikal= SmartTags("Sap Loggaus DB_Structuuri valvomolle.Aikaleima")
28 aika = CDate(LogattuAikal)
29 Aika4 = Right("0" & DatePart("yyyy",aika), 4)
30 Aika2 = Right("0" & DatePart("m",aika), 2)
31 Aika3 = Right("0" & DatePart("d",aika), 2)
32 Aika5 = Right("0" & DatePart("h",aika), 2)
33 Aika6 = Right("0" & DatePart("n",aika), 2)
34 Aika7 = Right("0" & DatePart("s",aika), 2)
35
36
37 Aikaa=Aika4&"-"& Aika2&"-"&Aika3&" "& Aika5&" ":"&Aika6&" ":"&Aika7
38
39
40
41
42 Lastausmaara= SmartTags("Sap Loggaus DB_Structuuri valvomolle.Lähtöpaino")
43 Tositenumero = SmartTags("Sap Loggaus DB_Structuuri valvomolle.Tositenumero")
44 RekNum = SmartTags("Sap Loggaus DB_Structuuri valvomolle.Tunniste")
45 Kuljetusliike = SmartTags("RFID apumuuttajat_Aktiivinen Rekka")
46 Asiakas = SmartTags("LastausMuisti_Rivi.Vastaanottaja_1")
47 Paikkakunta = SmartTags("LastausMuisti_Rivi.Vastaanottaja_3")
48 Numero= SmartTags ("LastausMuisti_Rivi.Numero")
49
50 Set yhteys = CreateObject("ADODB.Connection")
51 yhteys.ConnectionString = "DSN=ApexRAP"
52 yhteys.Open
53
54 Set objCommand = CreateObject("ADODB.Command") 'luodaan komento objekti
55
56
57 lisays = "INSERT INTO LastausPaivakirja VALUES ('" & Aikaa &"', '" & Lastausmaara &"', '" & Tositenumero &"', '" & RekNum &"',
58
59
60 Set objCommand = yhteys.Execute(lisays) 'tehdään lisäys
61
62
63 Set objCommand = Nothing 'vapautetaan komento obj
64 yhteys.Close 'suljetaan yhteys
65 Set yhteys = Nothing 'vapautetaan yhteys obj
66
67
68
69 End Sub

```

### Liite 3. Tulosta\_Rahtikirjat scriptti

```

1 Sub Tulosta_Rahtikirjat()
2 'Tip:
3 ' 1. Use the <CTRL+SPACE> or <CTRL+I> keystroke to open a list of all objects and functions
4 ' 2. Write the code using the HMIRuntime object.
5 ' Example: HmiRuntime.Screens("Screen_1").
6 ' 3. Use the <CTRL+J> keystroke to create an object reference.
7 'Write the code as of this position:
8
9 If SmartTags("TulostusNumero") < 5 Then
10     SmartTags("TulostusNumero")=SmartTags("TulostusNumero")+1
11
12
13     If SmartTags("TulostusNumero") = 1 Then
14         SmartTags("Numero")=SmartTags("Numero")+1
15         PrintReport("Rahtikirja_Tulostus")
16
17     End If
18
19     If SmartTags("TulostusNumero") = 2 Then
20
21         PrintReport("Rahtikirja_Tulostus")
22
23     End If
24
25     If SmartTags("TulostusNumero") = 3 Then
26
27         PrintReport("Rahtikirja_Tulostus")
28
29     End If
30
31     If SmartTags("TulostusNumero") = 4 Then
32
33         PrintReport("Rahtikirja_Tulostus")
34
35     End If
36
37     If SmartTags("TulostusNumero") = 5 Then
38
39         PrintReport("Rahtikirja_Tulostus")
40
41     End If
42
43 End If
44
45
46
47
48
49
50 End Sub

```

## Liite 4. Tagien hallinta ohje

### Henkilötunniste

Mene pääsivulta taagit välilehteen. Järjestelmä kysyy tässä vaiheessa käyttäjätunnusta ja salasanaa.



**kemira** PROSESSIKAAVIO PARAMETRIT TRENDIT HÄLYTYKSET KAMERAVALVONTA **TAAGIT** KIRJAUTUMINEN

Aktiivinen henkilö: Ei kirjautunutta käyttäjää  
Aktiivinen rekka: Ei kirjautunutta rekka-autoa  
Rekan rek.num. Ei kirjautunutta rekka-autoa

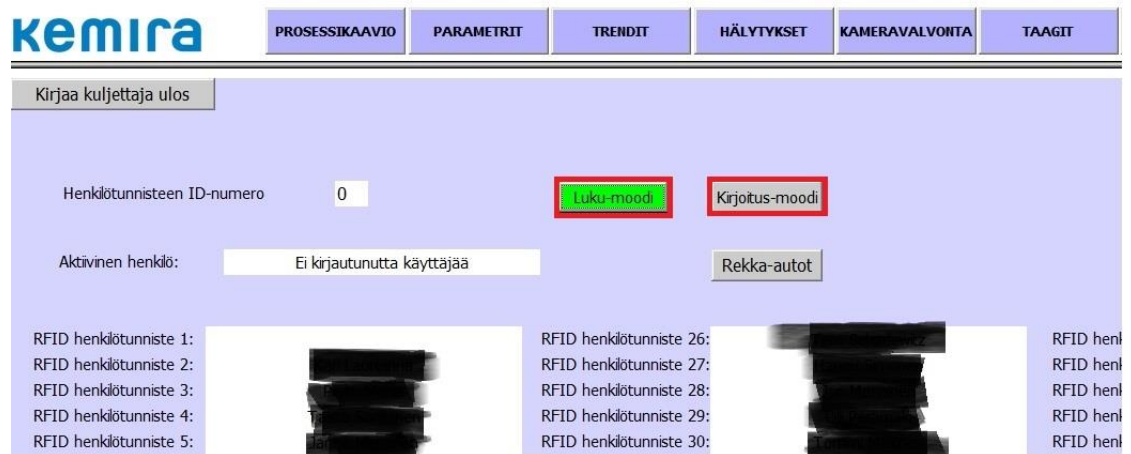
Asiakas:   
Vastaanottajan viite:

Lastattava määrä: +0,0 t  
Lastattu määrä osioon: +0,0 t  
Kokonaismäärä valitulle asiakkaalle: +0,0 t

Koko auto lastattu/  
Rahtikirjan tulostus

KUVA 1

Henkilökortin ID - numeron vaihtaminen tapahtuu tällä sivulla. Jotta kortille voidaan tallentaa ID - numero, täytyy luku-moodi vaihtaa kirjoitus-moodiin. Vihreä taustaväri kertoo kumpi moodi on käytössä (Kuva 2).



**kemira** PROSESSIKAAVIO PARAMETRIT TRENDIT HÄLYTYKSET KAMERAVALVONTA **TAAGIT**

Kirjaa kuljettaja ulos

Henkilötunnisteen ID-numero: 0

Aktiivinen henkilö: Ei kirjautunutta käyttäjää

RFID henkilötunniste 1:   
RFID henkilötunniste 2:   
RFID henkilötunniste 3:   
RFID henkilötunniste 4:   
RFID henkilötunniste 5:

RFID henkilötunniste 26:   
RFID henkilötunniste 27:   
RFID henkilötunniste 28:   
RFID henkilötunniste 29:   
RFID henkilötunniste 30:

**Luku-moodi** Kirjoitus-moodi

Rekka-autot

KUVA 2

Kun kirjoitus-moodi on käytössä, syötetään uusi ID - numero *Henkilötunnisteen ID-numero* kenttään ja painetaan enteriä (Kuva 3). Kun ID - numero on asetettu luetaan korttia lukijassa, jolloin ID - numero siirtyy korttiin.



KUVA 3

### Muista lopuksi vaihtaa luku-moodi takaisin käyttöön!

Tunnistetta voit testata painalla *Kirjaa kuljettaja ulos* painiketta, jolloin *henkilötunnusten ID-numero* nollaantuu ja *Aktiivinen henkilö* poistuu. Tämän jälkeen luotutetaan korttia lukijassa.

### Autojen tunnisteet

Autojen tunnisteita päästään hallitsemaan Rekka-autot painikkeesta.



KUVA 3

Jos halutaan vaihtaa autotunnisteen ID - numeroa, se täytyy ensiksi luotuttaa lukijassa. Luettu ID tulee näkyviin *Luettu ID* ja *Uusi ID* kenttään (kuva 4).





**kemira** PROSESSIKAAVIO PARAMETRIT TRENDIT HÄLYTYKSET KAMERAVALVONTA T

Luettu ID 037800040560004400000402 Yhdistä

Uusi ID 037800040560004400000402 Disconnect

Pysäytä lukeminen Tallenna uusi ID Henkilöt Kirjaa auto ulos

Aktiivinen rekka: Ei kirjautunutta rekka-autoa utunutta rekka

KUVA 4

Tämän jälkeen lukeminen pysäytetään *Pysäytä lukeminen* painikkeesta (painikkeen tausta muuttuu punaiseksi) ja annetaan *Uusi ID* kenttään uusi ID - numero ja paina enteriä. Tunniste sisältää 24 numeroa. Ensimmäiset 21 numeroa täytyy aina olla nollia ja viimeisiin kahteen numeroon annetaan vapaana oleva paikka (Kuva 5).



**kemira** PROSESSIKAAVIO PARAMETRIT TRENDIT HÄLYTYKSET KAMERAVALVONTA T

Luettu ID 000000000000000000000000042 Yhdistä

Uusi ID 000000000000000000000000042 Disconnect

Pysäytä lukeminen Tallenna uusi ID Henkilöt Kirjaa auto ulos

Aktiivinen rekka: Ei kirjautunutta rekka-autoa utunutta rekka

KUVA 5

ID - numeron tallennus tapahtuu *Tallenna uusi ID* painikkeesta. Tätä ennen kuitenkin tunnisteiden täytyy olla antennien lähettyvillä. Jos ID - numeron tallennus on onnistunut, uusi ID - numero tulee näkyviin *Luettu ID* kenttään (Kuva 6).



The screenshot shows the 'Kemira' software interface with a top navigation bar containing buttons for 'PROSESSIKAAVIO', 'PARAMETRIT', 'TRENDIT', 'HÄLYTYKSET', 'KAMERAVALVONTA', and 'T'. The main area has a light blue background. It features two input fields for IDs: 'Luettu ID' and 'Uusi ID', both containing the value '0000000000000000000042'. To the right of these fields are buttons 'Yhdistä' and 'Disconnect'. Below the 'Uusi ID' field are two buttons: 'Pysäytä lukeminen' (highlighted with a red box) and 'Tallenna uusi ID' (also highlighted with a red box). Further right are buttons 'Henkilöt' and 'Kirjaa auto ulos'. At the bottom, there is a section 'Aktiivinen rekka:' with a dropdown menu showing 'Ei kirjautunutta rekka-autoa' and 'utunutta rekka'.

KUVA 6

**Muista lopuksi käynnistää lukeminen *Pysäytä lukeminen* painikkeesta!**  
Tunnisteen voit testata painamalla *Kirjaa auto ulos*, jolloin *Luettu ID* ja *Uusi ID* kenttä nollaantuu. Tämän jälkeen voit käydä lukemassa tunnisteen normaalisti.

## Kuljettajan lisääminen

Samalla sivulla, jossa hallitaan henkilötunnisteita on myös listat kuljettajista. Kun järjestelmään lisätään uusi kuljettaja, kirjoitetaan kuljettajan nimi sille annetun tunnisteen ID – numeron kohdalla olevaan kenttään ja painetaan enter.

The screenshot shows the 'Kemira' software interface with a top navigation bar containing buttons for 'PROSESSIKAAVIO', 'PARAMETRIT', 'TRENDIT', 'HÄLYTYKSET', 'KAMERAVALVONTA', 'TAAGIT', and 'KIRJAUTUMINEN'. The main area has a light blue background. At the top left is a button 'Kirjaa kuljettaja ulos'. Below it, there is a section for 'Henkilötunnisteen ID-numero' with a dropdown menu showing '1'. To the right of this are buttons 'Luku-moodi' and 'Kirjoitus-moodi'. Below these are buttons 'Aktiivinen henkilö:' and 'Rekka-autot'. In the center, there is a text input field with a red box around it, containing the text 'ID - numero' and 'Tekstikenttä johon kuljettaja kirjoitetaan'. Below this input field is a list of RFID identifiers (RFID henkilötunniste 1: to 60:). The first identifier, 'RFID henkilötunniste 1:', is highlighted with a red box. To the right of the list, there are three columns of blurred images, likely representing RFID tags or drivers.

KUVA 7

## Kuljetusliikkeen lisääminen

Kuljetusliikkeen lisääminen tapahtuu samalla periaatteella kuin henkilön lisääminen.

Luettu ID: 000000000000000000000001

Uusi ID: 000000000000000000000001

Yhdistä

Disconnect

Pysäytä lukeminen

Tallenna uusi ID

Henkilöt

Kirjaa kuljettaja ulos

Aktiivinen rekka:

ID - Numero

Kuljetusliike kenttä

Rekisterinumero kenttä

RFID rekkatunniste 0001:

RFID rekkatunniste 0002:

RFID rekkatunniste 0003:

RFID rekkatunniste 0004:

RFID rekkatunniste 0005:

RFID rekkatunniste 0006:

RFID rekkatunniste 0031:

RFID rekkatunniste 0032:

RFID rekkatunniste 0033:

RFID rekkatunniste 0034:

RFID rekkatunniste 0035:

RFID rekkatunniste 0036:

KUVA 7

### HUOM!

Kuljettaja ja kuljetusliikkeen tietoja voi muutella milloin vain. Tietojen muutos ei vaadi sitä, että tunniste on kirjautuneena järjestelmään.